X-715-72-226
PREPRINT

NASA TM X- 65937

APPLICATION OF AN ONBOARD PROCESSOR TO THE OAO C SPACECRAFT

(NASA-TM-X-65937) APPLICATION OF AN ONBOARD PROCESSOR TO THE OAO C SPACECRAFT W.N. Stewart, et al (NASA) Jun. 1972 121 p CSCL 09B
N72-27211
Unclas
G3/08 34247

W. N. STEWART
R. HARTENSTEIN
C. TREVATHAN

JUNE 1972

GSFC
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-715-72-226

APPLICATION OF AN ONBOARD PROCESSOR TO THE OAO SPACECRAFT

W. N. Stewart
R. Hartenstein
C. Trevathan

June 1972

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

ABSTRACT

This document describes the design of a stored program computer for spacecraft use and its application on the fourth Orbiting Astronomical Observatory (OAO C). The computer, referred to as OBP 1, is a medium scale, parallel machine with a memory capacity of 16 384 words of 18 bits each. It possesses a comprehensive instruction repertoire and operates on 45 W of power (including the dc-to-dc converter). The machine operates at a 500-kHz rate and executes an add instruction in 10 μs. The primary functions of OBP 1 on OAO C will be auxiliary command storage, spacecraft monitoring and malfunction reporting, data compression and status summary, and possible performance of emergency corrective action for certain anomalous situations.

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

CONTENTS–Concluded

LIST OF ACRONYMS

A	Accumulator
ACMU	Auxiliary command memory unit
ADR	Address register
AIR	Allowable interrupt register
APH	A phase
BL	Block length
BPH	B phase
C	Carry (register)
CDC	Command data controller
CDH	Command data handler
CPU	Central processor unit
CRE	Command receiver equipment
CSC	Cycle-steal control
D	Decision (register)
DMA	Direct memory address
DTL	Diode transistor logic
EA	Extended accumulator
ELF	Enable lockout function
ETCU	Experimenter's test control unit
FDS	Flight Data Storage (Branch)
FWJC	Fine wheel and jet controller
GCWD	Gimbal command word
GFEI	Government furnished equipment interface
IC	Instruction counter
ID	Identification
I/O	Input/output
IOBB	Input/output unit output bus buffer
IRS	Instruction register
ISR	Interrupt storage register
LSB	Least significant bit
LSR	Lockout status register
LST	Lockout status table
MAB	Memory address line
MIB	Memory input data lines
MIBXXL	Memory input bus (number) lower
MIBXXU	Memory input bus (number) upper

MOB	Memory output data bus
MOB	Memory output data lines
MOBL	Lower memory output bus
MOBU	Upper memory output bus
MOBXXL	Memory output bus (number) lower
MOBXXU	Memory output bus (number) upper
MOR	Memory operand register
MSB	Most significant bit
OA	Or/and register
OAO	Orbiting Astronomical Observatory
OBP	Onboard Processor
OPL	Experiment command line
OV	Overflow register
P	Parity
PAGE	Page register
PC	Power converter
PCM	Pulse code modulated
PLM	Priority logic matrix
PPDS	Primary processor and data storage
SCALE	Scale register
S&C	Stabilization and control
SDHE	Spacecraft data handling equipment
SLF	Store lockout function
SLR	Storage limit (register)
SN	Serial number
SS	Subscript register
SSCU	Spacecraft systems controller
STSP	Star Tracker signal processor
WAITFF	Wait flip-flop

APPLICATION OF AN ONBOARD PROCESSOR TO THE OAO C SPACECRAFT

I. INTRODUCTION

This document describes the implementation of a computer for application on board the OAO C mission. The computer, developed at Goddard Space Flight Center (GSFC), was integrated with the spacecraft in February 1971. The primary goal of this new flight instrumentation and its associated ground equipment has been to improve the performance of the Orbiting Astronomical Observatory (OAO) C. In achieving this goal, most personnel involved in the project have gained the experience necessary to take full advantage of this powerful tool.

The Onboard Processor (OBP 1) interfaces with the rest of the spacecraft as shown in Figure 1 (DWG GE1308566).* This interface has been kept very simple and is configured in such a way that the OBP is not in line with any spacecraft subsystem. Hence, the OAO can operate in all normal modes with or without the computer. Primary functions of the computer on OAO C are auxiliary command storage, spacecraft status monitoring and reporting, limit checking, self-checking and diagnosis, and performance of emergency corrective action for certain anomalous situations. Secondary functions fall in the general category of circumventing single-point failure modes, whereby the computer serves as a backup in the event of failures in a number of spacecraft subsystems.

Experiment data processing will not be performed by the OBP on the OAO C mission, because the benefits to be gained by this function did not outweigh the interface complexity. Experiment data handling equipment already provides a very effective mode of operation which allows asynchronous sampling and storing of data under experiment control. However, it is felt that the OBP could benefit the experimenter on future spacecraft in the areas of data handling and data-dependent experiment control.

The intent of this document is fourfold: (1) to serve as an informal specification, (2) to provide interface facts required by GSFC and Grumman Aircraft Engineering Corporation spacecraft engineers, (3) to provide performance information required by operations personnel, and (4) to provide details of command and telemetry formats needed by the control center and other ground support equipment computer programmers.

*Number in parentheses refers to applicable OBP 1 logic drawing.

II. HISTORICAL DEVELOPMENT

The OBP was developed over a period of years by the Flight Data Storage (FDS) Branch. This development has included contracts with the Westinghouse Corporation for fabrication of an engineering model control processor unit (CPU), and with Electronic Memories Inc. for power-switched core memories. Along with these contracts, the FDS Branch developed the input/output (I/O) and power converter units, thus completing the OBP engineering model system. The OBP engineering unit was integrated as a system in October 1969 and was subjected to extensive thermal testing by the FDS Branch during November 1969 through January 1970.

The OBP flight model was fabricated under a contract with Micro-Technology Inc., which utilized a through-the-insulation weld technique developed by Micro-Technology.* This method of construction was used on both the CPU and I/O flight units, which were received at GSFC on March 1, 1970. The flight units were statically tested and ready for integration into the OBP flight system on April 5, 1970. The memory units, fabricated and tested by Electronic Memories Inc., had been delivered by February 1, 1970; and the power converter, designed and fabricated at GSFC by the FDS Branch, was tested and awaiting integration into the OBP system at this time. On April 5, 1970, all components of the OBP were integrated into the OBP flight system, and thermal testing began. The OBP system was subjected to many hours of thermal cycles and speed tests to verify the limits of operation.

During the temperature tests, problems with shorts occurring in the CPU caused some rework of the micro-point welding. This problem was traced to the method of terminating the micro-point wires and to the wire routing technique used on the CPU. The CPU, being the first flight unit constructed with micro-point techniques, was not wired with the improved routing techniques used in the I/O unit and was, therefore, finally replaced with a second CPU unit. This second unit, CPU 2, was constructed according to GSFC Specifications B-S-715-1 and B-S-715-3, which instructed the manufacturer of micro-point welded units in the proper techniques to ensure flight quality hardware.

The first flight acceptance test performed in accordance with the OAO C OBP qualification/acceptance test plan (see Appendix IV) was completed on August 31, 1970. During thermal vacuum testing, memory unit serial No. (SN) 3 and CPU 1 were not found to be wholly acceptable. The memory unit was returned to Electronic Memories Inc. and repaired. The OAO project at this time decided to replace CPU 1 as soon as possible, because of its history of shorts.

The OBP was first integrated with the OAO C spacecraft on December 14, 1970. Preliminary integration was successful, and no interface problems occurred. The OBP was configured with CPU 1 at this time.

For the final flight acceptance test, which was started on February 22, 1971, the OBP was configured with the CPU 2 unit and the repaired memory unit SN 3. On March 1, 1971,

---

*See "A Versatile Low Cost Packaging Technique for Spacecraft Electronics", GSFC Document X-715-70-333, Goddard Space Flight Center, Greenbelt, Maryland, March 1970.

the final OBP flight acceptance test was completed with no anomalies, and the OBP was delivered to the OAO Project for final integration with the OAO spacecraft.

The OAO flight software programs were used to verify that the hardware interfaces were functioning properly in a dynamic test begun on May 26, 1971, and completed on June 1, 1971. This included handling ACMU command transfers, loading, dumping, and changing specific locations in OBP memory. In addition, the power control and stabilization and control outputs of the OBP were functionally tested.

III. INTERFACE DEFINITION

General

The interfaces specified in this section were designed to minimize the impact on the normal spacecraft configuration while still allowing the OBP to demonstrate thoroughly its performance value both to the OAO C mission and to more advanced astronomical missions. To this end, all necessary electrical interfaces in the OAO C exist as originally configured. Other than cabling changes, essentially no spacecraft modification was required for computer integration. The interface also has been designed to minimize the effect of possible OBP failures on the normal operation of the spacecraft systems.

The interface connectors and circuits, shown in Figure 1, are in three major spacecraft subsystem areas: (1) command and timing, (2) telemetry, and (3) stabilization and control.

Command and Timing

The most significant OBP interface with the spacecraft is the interconnection of six relays between the Command Receiver Equipment (CRE) and the Primary Processor and Data Storage (PPDS). This relay configuration, the same as employed in the Auxiliary Command Memory Unit (ACMU) on OAO 2, provides a means for the computer to send commands to OAO subsystems via the PPDS with no modification to spacecraft hardware. This capability will greatly extend the usefulness of the OBP 1 to the mission. For instance, the task of using the computer to perform the function of the ACMU will be implemented. (More complex tasks that require this interface are discussed in Appendix A.)

These six relays are connected in such a way that the critical CRE-to-PPDS link is established with the relays in the released state. The relays are energized for a maximum of only 37 seconds and then are released by computer command. Four additional provisions for release of the relays are implemented as safety features; these are–

(1) Excluding the first 1-ppm signal following an energize command, all subsequent 1-ppm signals are gated to cause the release of the relays.

(2) Existence of command presence at the CRE overrides all OBP control and releases the relays.

(3) Disabling of primary power causes release of the relays.

(4) The relays and relay drivers are logically configured such that the CRE-to-PPDS connection is performed in spite of any single failure.

All ground control of the OBP is achieved via six impulse commands and two experimenter's command lines. The impulse commands provide absolute control of the computer and actuate critical on/off functions. These are power on/off, enable/disable OBP, and enable/disable OBP to Fine Wheel and Jet Controller (FWJC). All other functions are achieved by using the two experimenter's command lines. One line is used for loading of OBP programs, internal OBP commands, and the first half of the command pair when the ACMU function is performed. The second experimenter command line is used for loading the second half of the command pair for storage. The format for storing ACMU-type commands is essentially the same as that used for OAO 2. Format details are presented in Appendix B.

Other inputs from the PPDS are timing signals required for OBP operation.

Telemetry

The OBP receives all its spacecraft data through the serial data stream of the SDHE. No direct interface to any other equipment is required. This serial data stream and a frame synchronization pulse are taken from existing outputs of the Government Furnished Equipment Interface (GFEI) unit. The serial data are clocked into the OBP I/O unit by a 1042-bps signal derived from the PPDS.

The OBP has three types of outputs to the spacecraft telemetry system. One is 30 bits (two 15-bit words) in the SDHE main frame into which the OBP places various spacecraft status and performance code words and other low-frequency data. These 30 data bits are placed in the telemetry slots originally occupied by inner and outer gimbal command words for Star Tracker No. 1. Access to these slots is made by adding a gating circuit in series with the gimbal command word line which interconnects the PPDS and SDHE. To provide a continuous flow of gimbal command data to the telemetry even if the computer is disabled, this circuit is powered by the spacecraft's 18-V bus. This circuitry can be bypassed through a spacecraft signal controller unit (SSCU) relay. The second output is a direct, pulse code modulated (PCM), serial data stream to each wideband transmitter. This will normally be used to verify OBP memory contents, e.g., program or auxiliary command storage. At a bit rate of 50 kbs, any memory bank may be dumped twice in less than 7 seconds. The last type of data is 11 bi-levels for OBP status and voltage monitoring, and a single thermal-analog channel for temperature measurement.

Stabilization and Control

Only three connections are required to the stabilization and control (S&C) system to allow computer control of spacecraft attitude: These are the roll, pitch, and yaw analog error signals. All information required to compute these error signals is obtained from the SDHE data and requires no special interface. To implement this interface, digital signals are converted to analog and fed to the FWJC by way of the Star Tracker Signal Processor (STSP). Through use of one set of connections vacated by the omission of two star trackers, inputs

to this unit are made with no hardware modification. The analog voltage range is from −5 to +5 V from a 5000-Ω source.

Summary

The following lists summarize the OBP electrical interface requirements of the spacecraft system:

Commands

(1) Two experiment command assignments

(2) Six impulse spacecraft commands

(a) OBP power on

(b) OBP power off

(c) OBP enable

(d) OBP disable

(e) OBP-FWJC drive enable*

(f) OBP-FWJC drive disable*

Timing Signals From PPDS

(1) 1 ppm

(2) Command enable

(3) Bit time 3

(4) Bit time 32

(5) Clock time 1

(6) Clock time 2

(7) Clock time 3

(8) Clock time 4

(9) Star Tracker No. 1 address

(10) Gimbal command shift pulse

(11) Gimbal command word

*These commands exist as Star Tracker No. 1 inhibit signal to STSP.

Telemetry Input From SDHE

(1) SDHE serial data line

(2) Frame sync (bit 26, word 29)

(3) 1042 Hz

Telemetry Outputs

(1) 30 bits in the gimbal command word slot for Star Tracker No. 1 (one line)

(2) Dual input to wideband transmitters (two lines)

(3) 11 bi-levels and one thermal-analog channel to SDHE

Inputs From CRE

(1) Command message

(2) Message rate clock

(3) Carrier presence

Outputs to PPDS

(1) Command message

(2) Message rate clock

(3) Carrier presence

Outputs to STSP (Analog)

(1) Pitch error

(2) Roll error

(3) Yaw error

IV. OBP DESCRIPTION

General

The major design objectives in the development of the OBP system were high reliability, low power requirements, small size, and ease of programming. Reliability was emphasized in the design of the OBP at both the system and component levels.

At the system level, modular organization enables reliability enhancement through the ease with which redundant subsystems may be directly added, although for the OAO C instrumentation, no redundant subsystems are included. Even though there are no spare

modular units, the memory system is organized in such a way that any one of the four units can be switched by command to contain the interrupt control locations, thereby protecting against a worst-case memory failure.

At the component level, the following steps were taken to ensure maximum reliability of all subsystems:

(1) Maximum use was made of monolithic integrated circuits.

(2) To minimize failures due to electrical stresses, circuit components that operate well below rated values were chosen.

(3) Only those circuit techniques and components that have been tried and proven reliable were used.

(4) All active elements were "burned in" to minimize drift due to aging and to detect early failures.

(5) The number of electrical connections was minimized, and all internal connections were either welded or soldered.

(6) All electronic subassemblies were encapsulated.

(7) All fabrication was performed in a clean room.

This section will provide a summary of the significant features of the memory, the CPU, and the I/O subsystems. A more detailed discussion of each of these units is contained in Appendices A, C, and D.

Memory Subsystem

Briefly, the significant features of the memory subsystem are–

(1) Core memory with clear/write and read/restore operating modes.

(2) MIL Spec. version qualified on OAO 2 (ACMU subsystem).

(3) Operating temperature range of $-40°C$ to $+80°C$.

(4) Random access of 4096 words per unit with 18 bits per word.

(5) Four units for OAO C (16 384 words total).

(6) 2-μs cycle time.

(7) Operating power of 9-W average for program execution.

(8) Low standby power (less than 150 mW per unit).

Central Processor Unit

Significant features of the CPU are–

(1) Grammatically structured machine language.

(2) 18-bit instruction and data word size.

(3) 50 instructions, of which 30 require operand fetch.

(4) Binary 2's complement arithmetic.

(5) Fixed point with automatic scaling for multiply and divide.

(6) Hardware multiply and divide.

(7) Parallel data transfers.

(8) Single address one-word instructions.

(9) Low-power diode transistor logic (DTL) integrated circuit implementation.

(10) 10-μs add time; 68-μs multiply time; 140-μs divide time.

(11) 4-W power consumption.

(12) Stored program protection.

Input/Output Unit

Significant features of the standard I/O unit are–

(1) Two cycle-steal control channels (direct memory access).

(2) Both program and command initiation of cycle-steal channel.

(3) Redundant data bus in and out of memory (18 bits parallel).

(4) Eight interrupts with program-controlled priority levels.

(5) Low-power DTL integrated circuit implementation.

(6) 7.0-W power consumption.

Other significant components of the I/O subsystem for OAO C include the data channels necessary for throughputting data between spacecraft components and the OBP memory. The five peripheral circuits that are unique to the OAO C instrumentation are shown in Figure 2. These units are described in the following paragraphs.

SDHE Serial Data Buffer

The SDHE data arrive serially at a 1042-bps rate. These data, along with housekeeping data, form one 18-bit memory word. This word is fed to the OBP memory through input channel 2 at the 1042-bps rate via a cycle-steal channel (direct memory address). After each group of 26 words is input, a channel A block-length-equal-zero interrupt is generated. This interrupt serves as the real-time clock for the software. A frame rate signal (word 29, bit 26) presently used by the GFEI is used to initiate the SDHE buffer hardware and also interrupts the CPU for program synchronization.

Command Data Controller

The command data controller (CDC) performs the dual function of inputting commands from the PPDS (for OBP use) and outputting commands to the PPDS for their execution. Inputs to the CDC are 30-bit command words from experimenter command lines 1 and 2. Command bits 3 through 32 on both of these lines are gated into a common 30-bit shift register. These bits are transmitted in parallel to the memory input bus or to the cycle-steal control registers. Since the memory bus is an 18-bit interface, the 30 bits are split into two words of 18 bits and 12 bits. The 18-bit word is composed of bits 5 through 22 and enters the memory through input data channel 3. The 12-bit word is composed of bits 3, 4, and 23 through 32 and enters the memory through input data channel 10 octal. A detailed description of the command formats and the use of the command words is presented in Appendix B.

The output from the CDC is the 128-bit command format which is fed to the PPDS through the lines that are normally driven by the CRE. The relays shown in Figure 2 are energized to allow this transfer of command message. A complete command message is generated within the computer and output through data channel 1 as a serial bit stream. This message is shifted via a cycle-steal channel at a 1042-kbs rate.

Output Data Shift Register

The 30-bit output data shift register, composed of two 15-bit halves, serves as a parallel-to-serial converter for placing data into the SDHE bit stream. Data are loaded into each half of the register under program control through computer output channel 5. Since the 30 bits are fed into time slots previously occupied by Star Tracker No. 1 inner and outer gimbal command words, shifting is achieved by the gimbal command shift pulses. Shifting control is provided by the inner and outer gimbal select line, which determines the appropriate 15-bit portion to be output for a given period. As the serial data leave the register, they are circulated back to the input to form continuous bursts of output data which are or-gated into the gimbal command word line to the SDHE. The 30-bit register is reset by the computer at the SDHE frame rate.

Memory Dump Control

For memory dumps, blocks of up to 4096 words may be transmitted to ground by command control. The memory dump control (MDC) comprises a 32-bit shift register which serves as a parallel-to-serial converter to output data at a 50-kHz rate to the wideband transmitters. The 32-bit register is loaded with 18 bits of memory data through output data channel 0, and 12 bits of address from the appropriate cycle-steal control register. Another input to the MDC register is the 32-bit synchronization code, which occurs every 64 data words. As data are shifted out of the register, parity is determined, and a bit to form odd parity is inserted into the bit-32 time slot. The output of the MDC is a 65-word frame with the format depicted in Appendix B. Data dumps are achieved by the cycle-steal mode of

operation and can be initiated by the computer program or, in case of CPU failure or program error, by ground command to the MDC.

S&C Digital to Analog

The S&C interface supplies holding registers and three 6-bit (including sign bit) digital-to-analog converters for the three error output lines. The holding register is 18 bits long and is loaded by program control through output data channel 4. Each analog converter has an output range of -5 to $+5$ V and an output resistance of 5000 Ω.

Physical Description

The OBP is composed of seven separate boxes, packaged as shown in Figure 3. The total weight of the computer, excluding the mounting plate, is 64 pounds; a breakdown of the volume and weight of the major components is given below:

Unit	Volume (in.^{3})	Weight (lb)
CPU	270	14.75
I/O	300	16.25
Memories	500	23.5
PC	190	8.0
Cables	–	1.5

Three Deutsch connectors, specified in Figure 1 and designated AJ1, BJ2, and BJ3, are mated to spacecraft cables.

Power Consumption

The power requirements for the OBP in various operating modes are listed below. (Estimates of program execution time indicate that the computer will be in the halt state for a significant amount of time between real-time clock interrupts; this means the average operating power may be lower than indicated.)

+28 V (unregulated)	Operating Mode
17.7 W	standby
30.0 W	average operating
51 W	command transfer (for 37-s intervals)

+18 V (regulated)	Operating Mode
0.2 W	continuous

V. SOFTWARE SUMMARY

Executive Philosophy

The development and efficient checkout of those programs that are to fly on the OAO C mission are critical phases. Ease of programming was a prime consideration in the selection of the OBP instruction set and in the implementation of the OBP support software system.* In a real-time environment, however, problems often arise in the areas of time synchronization and input/output. In order to alleviate interface and timing problems for the OAO C mission, a real-time executive philosophy which eliminates some of the problems of applications programming was developed. The executive performs all input/output operations, including data decommutation and formatting; processes all interrupts; and monitors the execution of all applications programs with a time-shared, multipriority scheduling algorithm. Synchronization between the transfer of data into the OBP and the execution of program segments is derived from a real-time clock interrupt with a 26-ms period (SDHE data word rate). A new job cycle is initiated once each 1.59 s (an SDHE main frame) with an SDHE word-29, bit-26 interrupt.

Telemetry data from the SDHE are input along with OBP housekeeping data at a 1042-bps rate. As a result of the block-length-equal-zero pulse, a real-time clock interrupt request is generated for each 26-bit SDHE word. Following the repacking of data by the executive program, all SDHE data are stored in a 320-word spacecraft data buffer in the OBP memory. The commutation identification bits contained in the SDHE data are used to control the storage assignments of subcommutated data. Two words in the spacecraft data buffer contain system time and program code. Twenty-four buffer words are used for storage of Star Tracker gimbal command and error angles. Including all subcommutated levels, there are 15 SDHE digital words. One OBP buffer word is used to hold one-half of an SDHE digital word, and 30 buffer words contain all 15 SDHE digital words. The remaining 264 buffer words are used to store the 8-bit measurement of the 264 analog signals input to the SDHE.

When each SDHE main frame word is stored in the OBP memory (every real-time clock pulse) the executive program determines which applications program shall next receive control. Determination is based on both time and priority. A 16-word table is associated with each applications program. Three of the 16 words in each table indicate (1) which word number of a frame the worker wishes to begin computation, (2) the maximum computation time per frame for the worker, and (3) the priority of the worker. Thus, when each SDHE word is received, and also when an applications program has finished, the executive program determines if any new job requests are present and, if so, which is of highest priority. Control is passed to the job request having the highest priority, and the status of any program that may have been interrupted is saved so that control may be smoothly returned to the preempted program. A check is also made at each SDHE word time to determine whether the job in process has equaled its allotted computation time per frame; if so, the status of

*See "Support Software for the Space Electronics Branch On-Board Processor", GSFC Document X-562-68-388, Goddard Space Flight Center, Greenbelt, Maryland, November 1968.

this job is saved so that it may be resumed during the next frame. Other words in the 16-word table associated with each worker are used to hold the status of a worker if it is interrupted. Basically, then, workers are serviced on a priority basis after their starting times are reached; and, to eliminate a queuing problem from frame to frame, the total processing time for all jobs is kept less than 65 word intervals. The reason for giving a "time to start" option to the worker programs is to allow time-critical tasks to begin processing as soon as possible after the inputting of spacecraft data to be used by those tasks. The frame-sync interrupt is used to reset the real-time clock counter and thus maintain synchronization with the SDHE input data stream.

Interrupts

Eight interrupts are implemented for the OAO C mission. Interrupt 1 is the initiate interrupt and is unique in that it cannot be locked out; it is initiated either by ground command or by recovery from under voltage on board and is used to initiate certain data locations, clear off all interrupts, and pass control to the executive program. Interrupt 2, which is requested when either a command 2 or a command 1 with bit 32 equal zero is received, is used to store OAO commands (ACMU function). Interrupt 3, which results from an experiment command with octal 40 in bits 23 to 28, is used to send code words to the executive program. Interrupt 4 is SDHE frame sync (word 29, bit 26) and is used for synchronizing the inputting of SDHE data. Interrupts 6 and 7 correspond to cycle-steal channel A block-length-equal-zero and channel B block-length-equal-zero, respectively. These interrupts are requested after the last transfer to memory of a data block when the block length register has been decremented to zero; they signal that the block transfers are complete. Interrupt 6 is implemented as the real-time clock pulse, since it occurs every 26 ms while SDHE data are being input through cycle-steal channel A. Interrupt 5 is the outlimit interrupt and serves as a warning that memory storage is being attempted outside of established storage limits. Interrupt 8 is a time reference from the SDHE at the inner gimbal command rate.

VI. IMPACT ON GROUND OPERATIONS

General

The operation of the OBP will undoubtedly require an increase in OAO control center activity, especially in the command and data handling areas. It is important to note, however, that overall spacecraft operation can be significantly improved by the appropriate use of the OBP. As previously outlined, the OBP is capable of full-time, in-depth spacecraft monitoring. The results of this monitoring could provide a report of significant events and/or suspected or real problems to the ground either automatically or upon request.

The provisions for OBP command capability not only provide for the duplication of the ACMU function as flown on OAO 2 but also opens up a whole new area of highly sophisticated operational techniques that will have enormous impact on the present OAO operational scheme. For instance, the OBP could be programmed to make logical decisions

on input data and, based on the results, either perform experiments or take predefined courses of action. It is therefore possible to make the OAO self-diagnosing and self-adapting to any combination of measurable circumstances. This area requires much further study if it is to be fully utilized. Fortunately, the OBP software repertoire can be changed at any time, even after it is in orbit, in response either to unpredictable occurrences or to new requirements generated by the operations manager.

Ground Support Requirements

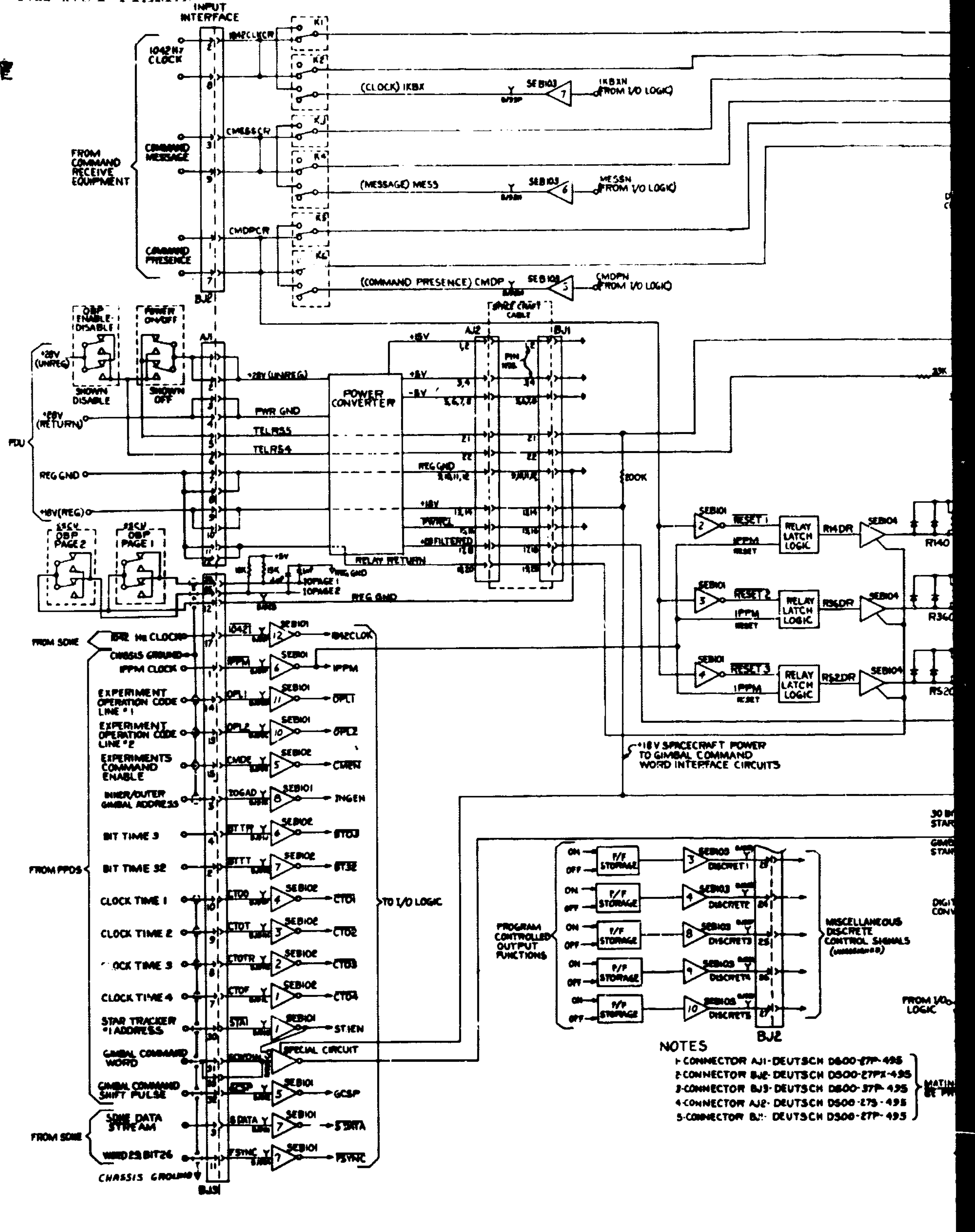
In addition to the impact on operational concepts and procedures, the OBP requires a significant computer capability within the control center. The most important of these requirements is the need of a computer to decipher and appropriately display the information contained in the two OBP words in the SDHE format. Interpretation of these words must be performed in real time if full advantage is to be taken of the subsystem status report feature. A second requirement is that of inputting a wideband dump of OBP memory into a ground-based computer and performing a bit-by-bit check on the array. This is necessary for validating new program loads and analyzing the results of certain diagnostic execution procedures.

Many other requirements on the control center computer will become apparent as OBP software progresses and operational philosophies develop. It is anticipated that many of the actual flight worker programs will be partially specified and designed by the operating personnel. This should circumvent incompatibilities between the onboard programs and the ground users.

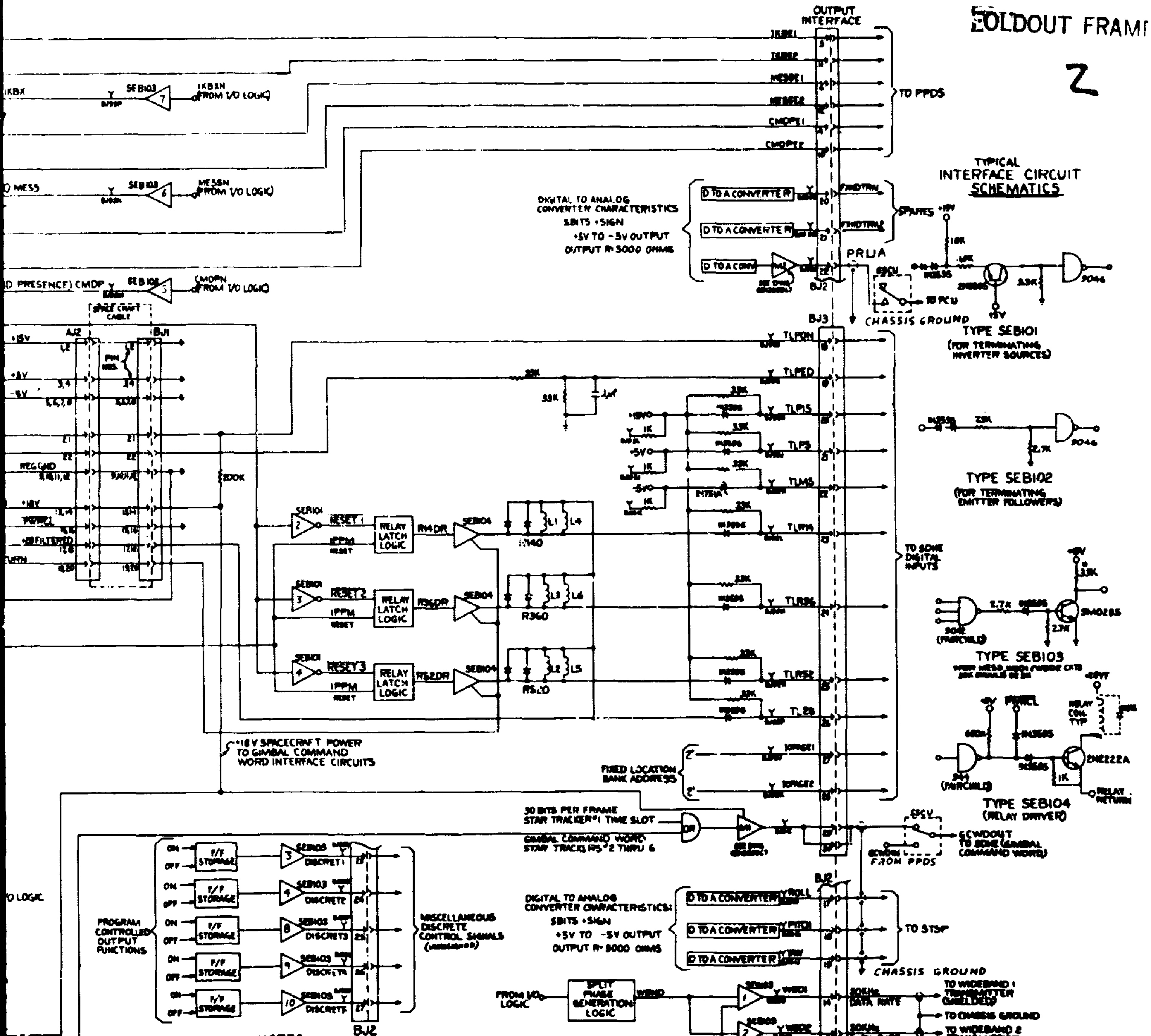
The software schedule included in this report covers only those efforts necessary to write and test OBP flight programs. The SDS-920 system at GSFC has been used to test all auxiliary command functions with a background of SDHE activity by using existing computer interface hardware and a PCM simulator. The proper operation of all applications programs are being verified by using the SDHE simulation program developed for the SDS-920 and SDS-9300 systems. A complete dynamic test of the OBP on the OAO C spacecraft is now underway and has met all major milestones necessary for spacecraft operations. So that proper OBP operation can be verified at a maximum of I/O activity and program execution complexity, auxiliary command functions are performed while subsystem equipment faults are tested and memory dumps are requested. Doing all of these things simultaneously requires a ground command and wideband telemetry capability to send commands to the OBP and verify memory dumps from the OBP. Provisions are being made to simulate equipment fault conditions in SDHE telemetry and process the SDHE data in the test facility to verify that the OBP has correctly reported these faults. An important segment of ground support software is being used to convert a set of 18-bit words into the experiment commands required to send these words into OBP memory so that the in-orbit reprogramming feature can be tested. It is worth noting that this ground support software development has two incidental benefits: It represents the bulk of the support software required for orbital operations, and it is being effectively used to aid in OAO subsystem integration in that the OBP is monitoring and reporting the status of subsystems in a manner that would please the most demanding diagnostician.

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME 1



FOLDOUT FRAME 2



NOTES

1-CONNECTOR AJ1-DEUTSCH D500-27P-495
2-CONNECTOR BJ2-DEUTSCH D500-27PX-495
3-CONNECTOR BJ3-DEUTSCH D500-37P-495
4-CONNECTOR AJ2-DEUTSCH D500-27S-495
5-CONNECTOR BJ1-DEUTSCH D500-27P-495

MATING CONNECTORS TO BE PROVIDED BY GAEC

Figure 1 (GE1308566)—OBP interface diagram.

PRECEDING PAGE BLANK NOT FILMED

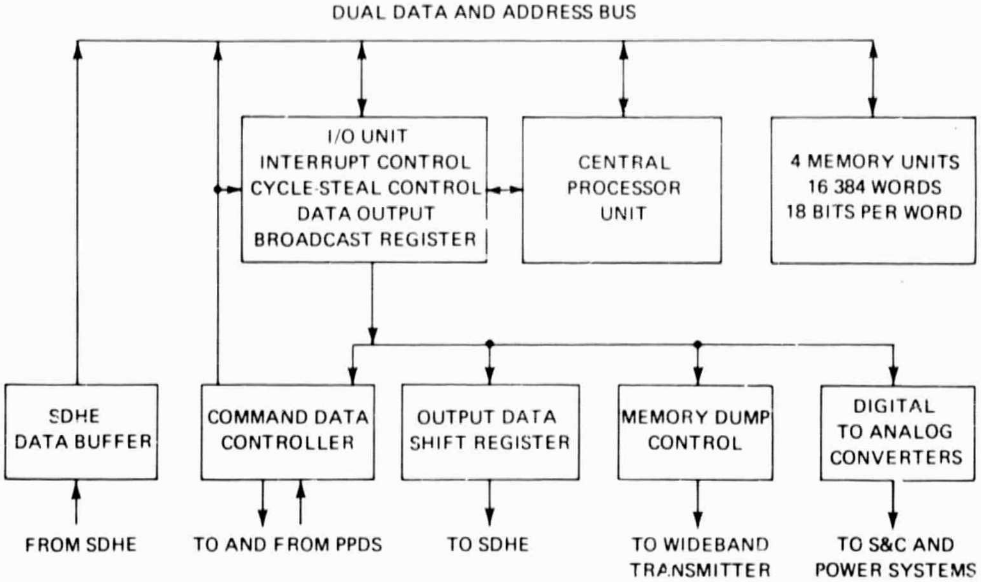


Figure 2—OBP block diagram.

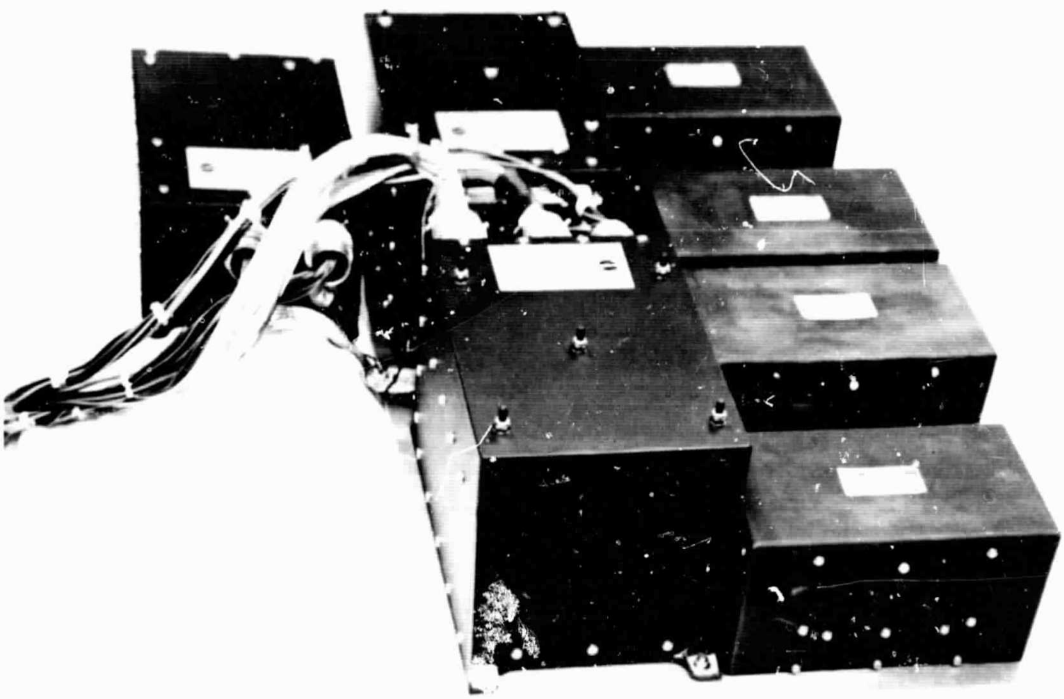


Figure 3—View of assembled OBP.

PRECEDING PAGE BLANK NOT FILMED

APPENDIX A

INPUT/OUTPUT UNIT DESCRIPTION

INTRODUCTION

A major objective in the design of the I/O unit for the OBP was to provide for multi-mission applicability. This requirement was met by defining an I/O interface unit that could be tailored to the specific mission requirement with minimum impact on the CPU or memory design. To enhance further the versatility of the I/O unit, a significant portion of it was designed so that for a variety of mission applications it would remain unchanged or require only minor modifications. This portion of the I/O unit includes–

(1) Two cycle-steal control (CSC) channels.

(2) Eight interrupt channels with lockout control.

(3) The I/O unit output bus buffer (IOBB).

(4) Ten data channels (five input and five output).

(5) The I/O unit control logic.

Also included with this group of the hardware, but functionally separate from the I/O unit, are the system clock and the memory access controller, referred to as the bus controller. The special section of the I/O unit that was designed for the OAO C application includes–

(1) The CDC.

(2) The MDC.

(3) The gimbal command word shift register.

(4) Six digital-to-analog converters.

A block diagram of the I/O subsystem is shown in Figure A1.

CYCLE-STEAL CONTROL

Cycle-steal operation is a method of transferring a block of data to or from memory independently of program execution. The CSC "channels", hereafter referred to as the CSC's, provide both memory addressing and read/write control to the memory. They also

handle the requesting of memory cycles through the bus controller (see "Bus Controller", below), which actually effects the interleaving of memory activity sometimes referred to as "cycle stealing".

Implementation

The OBP has two CSC channels: CSCA and CSCB. The two identical CSC's are completely interchangeable and thereby provide a degree of redundancy in the I/O unit. In addition, since they can operate simultaneously on different tasks, they provide added flexibility to the programmer.

Operation

Each CSC can be linked to any of the first four data devices. Using the request-acknowledge technique throughout the operation, the CSC maintains data communication at the device-dependent rate by interfacing the device and memory. Activity of the CSC is initiated, terminated, or reassigned by executing a *CONNECT TO* instruction or by a ground command via the Command Data Handler (CDH). Termination of the CSC activity also occurs when the block length (BL) has been decremented to zero. As each request from the external device is acknowledged, the CSC increments the starting address by one count and decrements the block length register by one count until BLA or BLB equals zero. At this point all further requests on that particular channel from its external device are ignored. This points out that a maximum block length (7777_8) will result in accessing only 4096_{10} rather than 4096_{10} memory locations. Accessing the remaining memory location is achieved by re-initiating the CSC with the appropriate address and block. As the block length register is decremented from one to zero, an interrupt is generated: interrupt 6, called $BLA = 0$ and associated with CSCA; and interrupt 7, called $BLB = 0$ and associated with CSCB. These interrupts signify to the CPU that the buffer transfer on the respective CSC channel has been completed. The CPU may then reassign the CSC to new tasks as the need arises.

Hardware Description

The two CSC channels in the OBP are identical and separate. Each contains a 16-bit starting address register and a 14-bit control register (Figures A2 through A5; GE1308724-27). Functionally, the control register has a 12-bit block length register and a two-bit device (data channel) register. The starting address register is a fully parallel, 16-bit up-counter; the block length register is a fully parallel down-counter; and the device register is a two-bit reset-set (R-S) latch register.

These registers are loaded by using a common clear and selective set technique. At the trailing edge of the appropriate CSC acknowledge, the starting address register is incremented by one, and the block length register is decremented by one until it equals zero.

CSC Initiation

CSC operation can be initiated in two ways: program control and command control.

Program control. Under program control, a CSC is loaded from the IOBB upon the execution of a *CONNECT TO (M)* instruction. This instruction causes the content of memory location *M* to be loaded into the control register and the content of the CPU accumulator to be loaded into the starting address register. During the *CONNECT TO* instruction, the CPU sends two control signals, called A phase (APH) and B phase (BPH), to the I/O control unit to indicate which of these two data transfers is in progress.

For purposes of the following discussion, refer to timing diagram of Figure A6. During the APH, data are transferred from a memory location to the IOBB. Sequentially in the time between APH and BPH, both the control register and starting address register of the particular CSC are cleared; then, the control register is loaded with data from the IOBB. Also during this interval, the CPU stores the content of the accumulator in location 7 of the fixed memory bank. During the BPH, the content of location 7 is read out of memory and placed in the IOBB; from there it is transferred to the starting address of the appropriate CSC. The CSC is now initiated and ready to honor data requests from the specified device. These requests are honored until the block length register equals zero.

Command control. Under command control, two commands must be sent consecutively to the command data handler to initiate a CSC channel. An experiment command (*FN CODE 01*) to load the control register must be sent first. This command clears both the starting address and control register of the specified CSC and then loads the control register. A second command (*FN CODE 02*) containing the starting address may be transmitted next. (Details of all command formats are discussed in Appendix B.)

CSC Status Monitoring

Once the CSC is in operation, its status can be continuously monitored under program control. The content of the block length register and the device number assigned to either CSC can be examined at any time by executing a data input instruction *LET INPUT FROM* on data device 6 for CSCA and on data device 7 for CSCB. This capability provides the programmer with all the information needed for adequate monitoring and control of the I/O unit activities.

Logic Reference

For complete details on logic implementation, refer to Figures A7 through A9 (GD1136184, GD1136182, and GD1136185).

INTERRUPT CONTROL

Introduction

As is the case with most general-purpose digital computers, OBP has a priority interrupt system. The interrupt control portion of the I/O unit stores and gates the individual interrupt, and in any given clock cycle allows only one interrupt to pass to the CPU for processing. These control functions are accomplished with an Interrupt Storage Register (ISR), a Lockout Status Register (LSR), an Allowable Interrupt Register (AIR), and a Priority Logic Matrix (PLM). A block diagram of this hardware is shown in Figure A10. The hardware priority logic prevents a conflict when several interrupts allowed by the AIR are waiting to be processed. The contents of the LSR are stored in the appropriate interrupt priority, fixed memory locations and thus provide programming control over the order in which interrupts will be processed.

OAO C Interrupts

For OAO C, eight interrupts were implemented; they are as follows:

(1) *INTR1 (INITIATE)*: Initiates program execution. A ground experiment command 1 with an octal code of 20 in bits 23 through 28 will generate this interrupt.

(2) *INTR2 (ACMU FN)*: Serves as an ACMU function interrupt used to store and transfer OAO commands. It is generated when a command 2 is received, or when a command 1 with bit 32 equal zero is received.

(3) *INTR3 (PROG CMD)*: Used to send a code word to the executive program. It is generated when an experiment command line 1 has an octal code of 40 in bits 23 through 28

(4) *INTR4 (FRAME SYNC)*: Used in a frame reference subroutine. It is generated when *FRAME SYNC* (word 26, bit 29) is received from the SDHE.

(5) *INTR5 (OUTLIMIT)*: Serves as a warning when the CPU tries to store in a restricted area of memory. It is generated by the *OUTLIMIT* signal from the CPU when an illegal write operation is attempted.

(6) *INTR6 (BLA = 0)*: Generated when the block length register of CSCA goes from one to zero.

(7) *INTR7 (BLB = 0)*: Generated when the block length register of CSCB goes from one to zero.

(8) *INTR8 (INGEN)*: Used as a constant time reference should the SDHE fail. It is generated at the trailing edge of the inner gimbal enable signal (*INGEN*).

Operation

An interrupt from a device arrives at the input of the ISR in the form of a 1.5- to 3-μs negative-going pulse. The stored interrupt is then compared with the corresponding bit

position of the LSR. If the LSR bit is a one, the interrupt is locked out; if the corresponding LSR bit is a zero, the interrupt will be clocked into the AIR. Since several allowable interrupts may exist in the AIR simultaneously, the priority logic must select the interrupt with the highest hardware priority. This interrupt level is then converted to a four-bit number and passed to the CPU. The interrupt number, also referred to as the interrupt address, is sent to the CPU via four interrupt address lines. When the CPU reaches the end of an instruction, or the *END INST* condition, it tests the interrupt address lines for the nonzero state on all four lines and recognizes this as an interrupt request. As the CPU proceeds to service the interrupt, it sends an acknowledge signal (*INTACK–*) back to the I/O unit so that the serviced interrupt may be cleared from the SIR.

<u>Lockout Status Table</u>

Hardware priority on OBP is set with interrupt one having the highest priority and interrupt eight having the lowest priority. The sole purpose of the hardware priority is to break ties when two or more interrupts happen in the AIR. By selective lockout, however, the real order in which interrupts are serviced can be programmed. All lockout words as a group are referred to as the Lockout Status Table (LST). If a simple priority order is assigned to the various lockout words, the resulting LST exhibits the shape of a pyramid as a minimum. As an illustration, consider the eight interrupts implemented for OAO C as having a lockout priority shown in Table A1.

Table A1–Sample lockout status table.

Interrupt	Hardware Priority	Interrupt Lockout Status Words								Address
		Bit								
		1	2	3	4	5	6	7	8	
INITIATE	1	1	1	1	1	1	1	1	1	14
ACMU FN	2		1	1		1			1	24
PROG CMD	3			1		1			1	34
FRAME SYNC	4		1	1	1	1	1	1	1	44
OUTLIMIT	5					1			1	54
BLA = 0	6		1	1		1	1		1	64
BLB = 0	7		1	1		1	1	1	1	74
INGEN	8								1	104

The order in which the interrupts will be serviced can be easily determined if one rearranges the interrupts such that the number of ones in each column increases as the service priority decreases. This is shown in Table A2.

Table A2–Rearranged lockout status table.

Interrupt	Hardware Priority	Service Priority	Interrupt Lockout Status Words							
			Bit							
			1	4	7	6	2	3	8	5
INITIATE	1	1	1	1	1	1	1	1	1	1
FRAME SYNC	4	2		1	1	1	1	1	1	1
BLB = 0	7	3			1	1	1	1	1	1
BLA = 0	6	4				1	1	1	1	1
ACMU FN	2	5					1	1	1	1
PROG CMD	3	6						1	1	1
INGEN	8	7							1	1
OUTLIMIT	5	8								1

First, from Table A2 it can be noted that the interrupt lockout table exhibits a priority service that is at least pyramid. Second, if an interrupt *X* with low service priority has to lock out an interrupt *Y* with a higher service priority, the lockout status table has to have *one* of the following two properties:

(1) All interrupts with higher service priority than *X* must be locked out.

(2) All interrupts with greater service priority than *X* must also lock out interrupt *Y*.

Table A3 illustrates the two properties when *X* equals the service priority of 7 and *Y* equals the service priority of 3.

In assigning the lockout status for an interrupt, it must be kept in mind that an interrupt with a high service priority but with a low hardware priority can experience a delay. This would happen if while servicing an interrupt of low hardware priority (i.e., *OUTLIMIT*), several interrupts of higher service priority occurred (i.e., *ACMU FN, BLA = 0, BLB = 0*). The computer would service at least one instruction of the *ACMU FN* and *BLA = 0* interrupts before servicing *BLB = 0*, which has a higher service priority than *ACMU FN* and *BLA = 0* interrupts.

There will be slight delays when a low service priority interrupt is set while servicing a high priority interrupt. At times, however, this delay can be used to advantage. Assume that *OUTLIMIT* is set while servicing *BLA = 0* interrupt. From Table A2, it is obvious that *OUTLIMIT* will be serviced only after honoring the higher priority interrupts *ACMU FN* and *PROG CMD* interrupts. The delay, however, can be used to hold off some non-time-critical task that is dependent upon *BLA = 0* but also wants to be at the first level of interrupt.

Table A3–Property of lockout status table for interrupt 8 to lockout interrupt 7.

Interrupts	Hardware Property	Service Priority	Interrupt Lockout Status Words*							
			Bit							
			1	4	7	6	2	3	8	5
INITIATE	1	1	0	1	1	1	1	1	1	1
FRAME SYNC	4	2		1	1	1	1	1	1	1
BLB = 0	7	3			1	1	1	1	1	1
BLA = 0	6	4			W	1	1	1	1	1
ACMU FN	2	5			W		1	1	1	1
PROG CMD	3	6			W			1	1	1
INGEN	8	7			WZ	Z	Z	Z	1	1
OUTLIMIT	5	8								1

*Property 1 = Z; property 2 = W.

Hardware Implementation of the I/O Unit

Lockout status register. The LSR is an 8-bit asynchronous R-S flip-flop register that holds the lockout status word of the interrupt. It is one of the nonaddressable registers whose content is exchanged whenever the CPU processes an interrupt or executes a *RESUME FROM* or *EXIT* instruction. The register is loaded with the content of fixed locations in memory (see Table A2) directly from the memory output bus bits 1 through 8. The register is loaded by a strobe pulse generated by a Read Complete signal from memory and an enable signal from the CPU (ELF). If address bit 13 is a one, the register is loaded from data in the upper memory output bus (MOBU); if address bit 13 is a zero, the register is loaded from the lower memory output bus (MOBL).

The output of the register is gated with the output of the ISR to determine the allowable interrupts. In addition, it is also stored in memory. This occurs when a CPU memory access acknowledge (*CPUACK*) and a store lockout function (SLF) are generated by the CPU.

Interrupt storage register. The ISR is an 8-bit, randomly set R-S flip-flop register that stores the occurrence of each interrupt until the interrupt is serviced. It is loaded by negative-going pulses (1.5- to 3.0-μs) and is cleared with a negative-going Power Clear signal from the power converter (PC). The individual flip-flop is also cleared whenever the individual interrupt is honored by the CPU.

The output of the ISR is and-gated bit by bit with the output of the LSR to determine the allowable interrupts.

Allowable interrupt register. The AIR is an 8-bit register used to synchronize the allowable interrupts with *PH PROC*. *PH PROC* is and-gated with an *INT–* signal from the CPU to effect a clock for the register that is active only when the CPU is not processing an interrupt.

This ensures that the interrupt address does not change in the middle of an interrupt processing; it also provides an updated set of interrupts at the end of every instruction cycle. The input to the register is from both the LSR and ISR; the output is fed to the priority matrix logic.

Priority logic matrix. Briefly, the PLM gate network establishes a hardware priority to break ties in the event that two or more interrupts are allowed at the same time. The complement of a high hardware priority interrupt is used to inhibit allowable interrupts of lower priority. The output of the PLM is fed to the encoding gates and then sent to the CPU as the interrupt address.

I/O BUS BUFFER REGISTER

Description

The IOBB, an 18-bit register, is the interface between the Memory Output Data Bus (MOB) and the I/O unit. The function of the register is to store the I/O control words and all output data temporarily. As applicable, the outputs are directed to the I/O unit control logic, the CSC channels, and all output data devices.

Operation

Prior to any load, the IOBB is always cleared. An IOBB load enable signal (*IOBB LDEN*) and a memory initiate (*MCLK*) signal are and-gated to generate an IOBB clear pulse (*BBCL–*). The load enable signal (*IOBB LDEN*) is also gated with a Read Complete signal from memory (*UBRC–* or *LBRC–*) to generate a 350-ns strobe pulse which enables the transfer of data from either output bus to the IOBB register. In order to give initial transients time to settle, the strobe pulse is delayed about 200 ns after the data are laid on the bus by the memory. The output of the IOBB is broadcast to all output devices; that is, any device using the IOBB data must gate them with the appropriate enable pulse.

Hardware Description

For the purpose of the following discussion, refer to Figure A11 (GD1308695). The IOBB register is an asynchronous register with 18 R-S latch flip-flops. The use of the latch flip-flops affords fast as well as asynchronous operation. Each flip-flop is loaded either from the upper or the lower output data bus, depending upon whether an even- or odd-numbered 4K memory bank was addressed.

Strobe Circuits

The strobing circuits of both the IOBB and the LSR use two interconnected latches to regenerate a clean strobe pulse when a Read Complete (*BRC–*) signal is received from memory. [The circuit is shown in Figure A12 (GD1308576).] The *BRC–* pulse must be present long enough for the first latch to be set.

OBP BUS SYSTEM

Description

The bus system interconnects the I/O unit, the CPU, and the memories address and data lines. All interconnecting buses are redundant. There is a lower bus to interconnect the logic with the even-numbered memory units and an upper bus to interconnect with the odd-numbered memory units. Each bus has four sections: memory address lines (MAB's), memory output data lines (MOB's), memory input data lines (MIB's), and control and power lines.

Address Lines

Each address cable has both the address lines and the power and control lines. An address cable has 16 lines for addressing purposes, three lines for control, seven lines for +5 V, seven lines for −5 V, three lines for +15 V, and six lines for ground. Since there are three signal sources to the address bus (CSCA, CSCB, and CPU), the address and read/write control lines must be interconnected to perform the "logical or" function. This is done by connecting collectors of normally off transistors and holding these lines "high" (logic one state) by the use of 5100-Ω pull-up resistors connected to +5 V in the I/O unit. With this scheme, the address lines are energized to the logic zero state by the turn-on of any source transistor. The read/write control line is pulled to ground by the appropriate source for a write operation. The other two control signals are Memory Initiate (*MINIT*), which is generated by the bus controller, and Read Complete (*BRC−*), which is generated by the appropriate memory unit. In the case of *MINIT*, the only source is the bus controller, and the "or" function is not required; therefore, this signal is normally at ground and goes positive. This is a significant factor in that ground on the *MINIT* line prevents undesired memory operations during OBP turn-on or turn-off.

The line driver for the address and the read/write line is a low-power, collector gate capable of sinking 6 mA. To speed the address line's return to +5 V (normal state), a 300-ns positive burst pulse is applied to the address bus at the end of each memory cycle. The Read Complete control signal, like the output data lines discussed below, is terminated by a 1000-Ω resistor to +5 V at the I/O end, but this one line also has added to it a 1000-Ω resistor to +5 V at the CPU end. The memory initiate control line (*MINITL, MINITU*) is normally low and is pulsed by a 500-ns pulse which initiates a memory cycle.

Output data bus. There are 36 data lines in the output data bus: 18 lines from the even-numbered memory units, and 18 lines from the odd-numbered memory units. The lines from the even-numbered memory units are referred to as the Memory Output Bus Lower (MOBXXL). The lines from the odd-numbered memory units are referred to as the Memory Output Bus Upper (MOBXXU). As discussed for the address bus, these lines are held normally high and are driven to ground when a zero is put on the line. A 1000-Ω resistor to +5 V terminates the lines at the I/O unit.

Input data bus. Like the output data bus, the memory input bus is made up of 36 data lines. Because each memory word is 18 bits long, only 18 of the lines go to each memory unit. Those lines going to the even-numbered memory units are referred to as the Memory Input Bus Lower (MIBXXL). Those going to the odd-numbered memory units are referred to as the Memory Input Bus Upper (MIBXXU).

Each line is held normally to a logical one (+5 V) by a 5000-Ω resistor to +5 V at the I/O unit. A line is pulled to ground whenever a logical zero is placed on that line. A low-power, open collector gate capable of sinking 6 mA is used as the line driver. To speed the data lines return to their normal state, a 300-ns positive burst pulse is applied to the bus at the end of each memory cycle.

OBP SYSTEM CLOCK

The OBP clock is a simple two-phase clock system with each phase having a 25-percent duty cycle and a 180-degree relative phase difference. A 1.0-MHz squarewave generated by a three-gate crystal-controlled oscillator [Figure A13 (GD1136183)] is divided into two 0.500-MHz timing signals which are gated with the 1.0-MHz wave to generate the two clock signals *PROCCLOCK* and *SYNCCL*. In both the CPU and the I/O unit, these signals are gated through two additional logic levels to provide drive capability and preserve the original timing in both units. They become *PHPROC* and *PHSYNC* in the I/O unit and *PROCCLOCK* and *SYNCCLOCK* in the CPU. The basic frequency of the oscillator is determined by the crystal. The clock pulse width was based on the clock response time of the 9040 bistable multivibrator. The clock rate was selected as a function of the CPU carry propagation time.

BUS CONTROLLER

The function of the bus controller logic is to supervise the activity of the memory system (see Figure A7). It establishes priorities and synchronizes requests made by CSCA, CSCB, or the CPU for memory access. When any two devices make simultaneous requests, the order of priority is CSCA, CSCB, and CPU. However, if simultaneous memory requests are made by all three, the order of service is CSCA, CPU, and CSCB; therefore, the CPU is guaranteed at least half the memory cycles.

In order to prevent a short or an open circuit on a CSC request line from capturing all memory cycles, the logic requires that a request be dropped after each memory cycle and be raised again for a minimum of one clock cycle (2.0 μs, in our case) prior to honoring a second memory cycle. Request on the CSC channels are synchronized on the leading edge of *PHSYNC* and are acknowledged at the trailing edge of *PHPROC*. A block-length-equal-zero on the particular CSC prevents additional memory requests from being honored. CSC requests are also inhibited during the execution of an I/O instruction by the CPU so that the *CONNECT TO* instruction can be properly executed on an active CSC.

The outputs of the bus controller are three memory acknowledge signals (*CCAACK−*, *CCBACK−*, *CPUACK−*), a 300-μs burst signal (*BURST−*), and three identical 500-ns memory initiate pulses. Two of the memory initiate pulses (*MINITL* and *MINITU*) are sent to

the memories, and the third (*MCLK*) is sent to the CPU for gating purposes. The *BURST* pulse is used to charge the input and address buses to +5 V at the end of a memory cycle. For complete timing of these signals refer to Figure A14 (GD1136189).

COMMAND DATA HANDLER

Introduction

The purpose of the Command Data Handler (CDH) is twofold: (1) to input commands from the PPDS, and (2) to output commands to the PPDS for execution.

The inputs to the CDH are 30-bit command words sent over experiment command line 1 or experiment command line 2. The data from both lines are gated with the command enable signal and shifted into a common 30-bit register. These commands are transmitted in parallel to the memory input bus or to the cycle-steal registers as applicable. Because the memory bus is an 18-bit interface, the 30-bit command words are split into two words of 18 bits and 12 bits. The 18-bit word comprises bits 5 (MSB) through bit 22 (LSB) and enters memory through input data device 3. The 12-bit word comprises bits 3, 4, and bit 23 through bit 32. This word is transmitted only to memory and is input through device 10. The details of the command formats and use of the command words are presented in Appendix B.

The output of the CDH is an OAO serial command that is transmitted to the PPDS through three lines: Command Presence (*CMDP*), Message (*MESS*), and 1-Kilobit Transfer Clock (*IKBX*). A complete 128-bit command message is generated within the computer and is output by a cycle-steal channel one bit at a time under software control through data device 1.

Hardware Description

The CDH comprises the command shift register [Figure A15 (GD1308597)], the command decoding and control logic [Figure A16 (GD1308599)], the relay enable logic [Figure A17 (GD1308602)], and the command transfer logic [Figure A18 (GD1308589)].

Command register. The command register is a 30-bit serial-to-parallel shift register that inputs data from experiment command line 1 (OPL1) or line 2 (OPL2) at a 50-kHz rate. The data are and-gated with the command enable signal (*CMDEN*) and shifted into the register at the trailing edge of clock time 2 (*CT02*). Bit time 3 (*BT03*) and bit time 32 (*BT32*) are gated with *CMDEN* to determine the start and stop of shifting the commands into the register. The output of the command register is transmitted in parallel to the memory or to the control channels as determined by the decoded commands.

Command decoding and control. The function of the decoding and control logic is to decode the incoming commands and generate the proper control signals. Bit 3 and bit 32 of an experiment command word are used to determine the type of command. Bit 3 is always a one for commands received on OPL1. Bit 32 received on OPL1 is used to distinguish

between the first half of an ACMU command and all other OBP commands. A one in bit 3 and a zero in bit 32 received on OPL1 indicates that bit 4 through bit 31 of the command are the 28 bits of the first half of an *ACMU FN* command (see ACMU command format in Appendix B). As a result of this condition, the Wait flip-flop (WAITFF) is set to a one state, and interrupt 2 (*INTR2*) is generated. The second half of an ACMU command must follow the first half because any other experiment command following the first half will reset the WAITFF and be treated as the second half of the ACMU command. When the second half of the ACMU command is received properly, WAITFF is reset and another *INTR2* is generated. Any experiment command is detected as being the second half of an ACMU command whenever WAITFF is a one and bit 3 from OPL1 is a zero. If bit 3 and bit 32 from OPL1 are both ones, bits 23 through 28 of the command are considered to be a 6-bit "hardware" function code determining an OBP command.

Relay enable logic. The function of the relay enable logic is to enable and disable the command transfer relays. Each relay driver consists of an R-S latch coupled to a relay driver circuit. A special flip-flop that disables the latch 1 to 2 minutes after it is set serves as a backup to prevent lockout of ground commands. Prior to any command transfer, the relays are enabled under software control. At the end of a command transfer, the relays are disabled also under software control. In the event that they are not disabled, the 1-ppm signal disables the relays within 1 to 2 minutes after they are set. A master clear command will also disable the relays. In the event that a ground command is sent, the relays are reset. [See Figure A17 and Figure 1 (in text).]

Command transfer control. For command transfer, only three master/slave flip-flops are used. Basically, they serve as 1-bit buffers for each of the three signals associated with command transfer. The entire command message is assembled within the computer and is output under CSC control to device 1 at a 1042-Hz rate. Once the command message is assembled in memory, the 1042 clock generates a data request.

MEMORY DUMP HANDLER

Introduction

The purpose of the MDH logic is to provide a memory dump capability under CSC control that can be initiated either by command or by the computer program. In both cases, dumps occur in blocks of up to 4096 memory words. When a dump is initiated by the program, the particular block of words is dumped only once, unless it is re-initiated. However, when the initiation is by command, not only does the dump start with the first word of the 4K memory bank selected, but the block of words is dumped twice. Proper control is provided by the dump control logic [Figure A19 (GD1308600) and Figure A20 (GD1308598)], and data are output on data device zero through a 32-bit parallel-to-serial shift register [Figure A21 (GD1308595)].

Dump shift register. The shift register is loaded in parallel with 18 bits of memory data, 12 bits of address (bit 3 through bit 14 of the appropriate CSC), one bit for registration purposes, and one bit for parity. The odd parity bit is added as bit 32 of each word as the data are shifted out of the register. The first word of each 65-word frame is a 32-bit sync code. For details of word structure and frame format, see "Dump Format" in Appendix B. The register is shifted at a 50-kHz rate, and its output is converted to split phase prior to transmission to the wideband transmitters.

Dump control logic. The function of the dump control logic is to generate the control signals for the proper data dump transmission. The normal dump format for a 4096-word block is 32 bits per word, 65 words per frame, 64 frames per dump. When dump is initiated either by command or by program, it is synchronized with *BT32*. The shift register is not cleared but is loaded with ones at the bits corresponding to ones in the frame sync pattern. This unknown data pattern is shifted out first and helps the receiver lock to the signal. On the next *BT32*, the proper frame sync pattern is loaded into the shift register, and the dump operation begins. If dump was initiated by program, it is terminated when the block length of the particular CSC equals zero; if dump was initiated by command, it is terminated by the command dump mode clear signal (*CMDDMPCL−*), which occurs at the end of the second dump.

Data request and dump register load are done during *BT32*. During this time, clock time 1 (*CT01*) and clock time 2 (*CT02*) are used sequentially to clear (*CLRDMPR−*) and then load the dump register with the appropriate address. Also, *CT02* is used to generate data request. The data are loaded into the register by data device zero enable pulse (*DC0EN*). When not in *BT32*, the register is shifted at the trailing edge of *CT03*. During *BT32*, an odd parity bit is added to the data stream, and the dump register is updated. For timing details refer to Figure A22 (GD1308660).

Command dump logic. When a dump is initiated by command, only the CSC channel identification (ID), the CSC block length, and the memory bank ID are sent as a command. As a consequence, when dump is initiated this way, it always starts with the first word of the selected memory bank. Another consequence is that the last word in the block cannot be read out of memory because the block-length-equal-zero condition inhibits the bus controller from honoring the 4096 memory requests.

The command dump control logic avoids this impasse by momentarily clocking the block length out of the zero condition. This allows full dump capability without impacting the design of the bus controller. For redundancy purposes, the selected block is also dumped twice.

When a dump command is received, as part of the dump initiation, the command dump mode flip-flop (*CMDDMPM*) is set. A one in this flip-flop causes the block length control register of the selected CSC to be decremented to the 7777_8 state from the all-zero condition. This allows the bus controller to honor memory requests. After one additional request is

honored, the CSC is cleared and reloaded from the command register. This allows the block of words to be dumped again. A one-bit counter (*DMP1*) is used to keep track of the number of times the block has been dumped; and after the second time, the dump mode is disabled. For a complete timing detail, refer to Figure A23 (GD1308658).

GIMBAL WORD SHIFT REGISTER

The gimbal word shift register was developed to allow the OBP data to be inserted as Star Tracker No. 1 data. Should a failure occur in the OBP, this interface can be bypassed by a ground command to avoid interference with the normal mode of SDHE data.

The 30-bit register comprises two 15-bit halves and serves as a parallel-to-serial converter for placing data into the SDHE bit stream. Data are loaded in parallel into each half of the register under program control through data output device 5. Since the 30 bits are fed into time slots previously occupied by the OAO Star Tracker No. 1 Inner and Outer Gimbal Command Word, shifting is achieved by the clock formed by the leading edge of the Gimbal Command Shift Pulses (*GCSP*) and the leading edge of *CT01*. Shifting control is provided by the Inner-Outer Gimbal Select line (*INGEN*), which determines the appropriate 15-bit portion to be output for the given period. As the serial data leave the register, they are circulated back to the input to form continuous bursts of output data. The data are complemented and or-gated into the Gimbal Command Word (GCWD) line to the SDHE. Each of the two 15-bit half-registers is independently updated under software control at the SDHE frame rate. Bit 18 of the update word determines whether the inner or outer gimbal half of the register is to be refreshed: If bit 18 is a one, the inner gimbal half is loaded; if bit 18 is a zero, the outer gimbal half is loaded. When the particular half of the register is updated, shifting for that half of the register is stopped until the proper time slot for the data is reached.

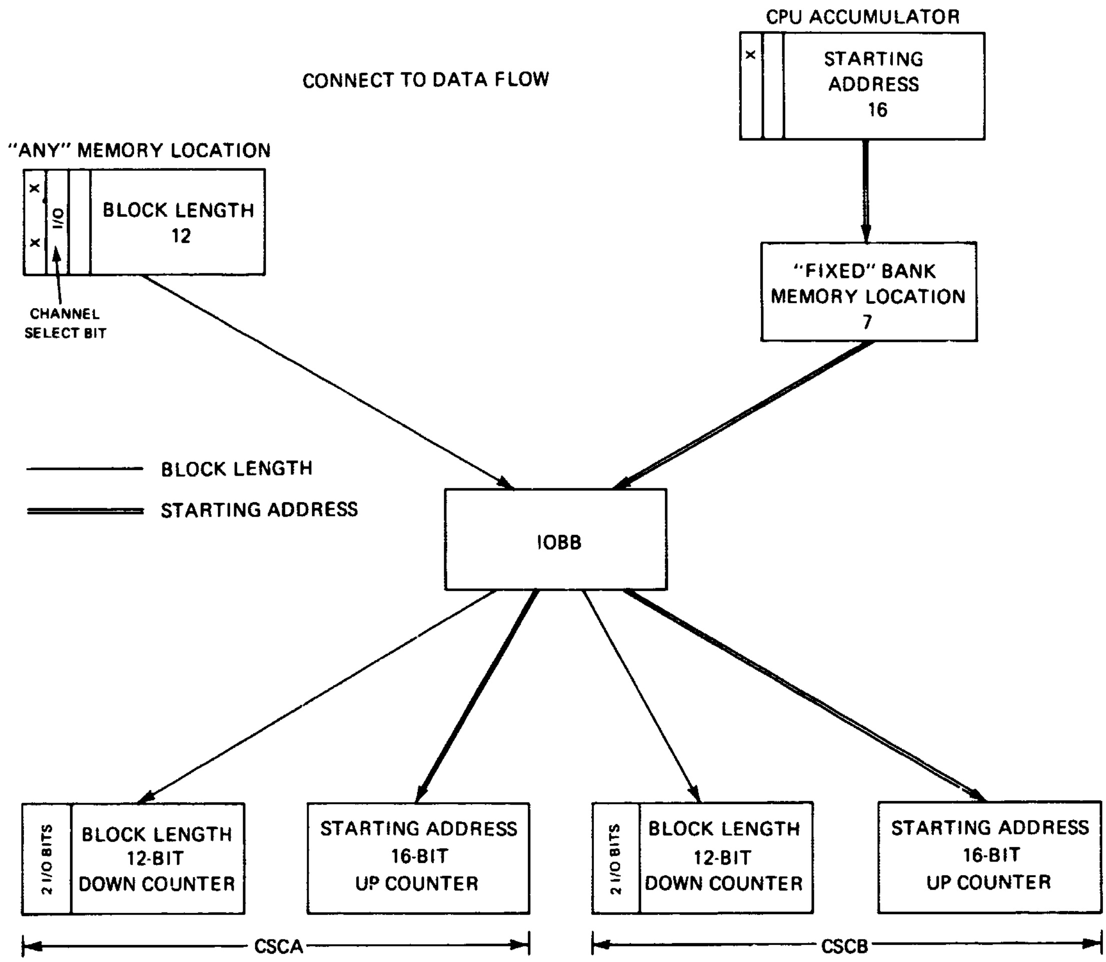
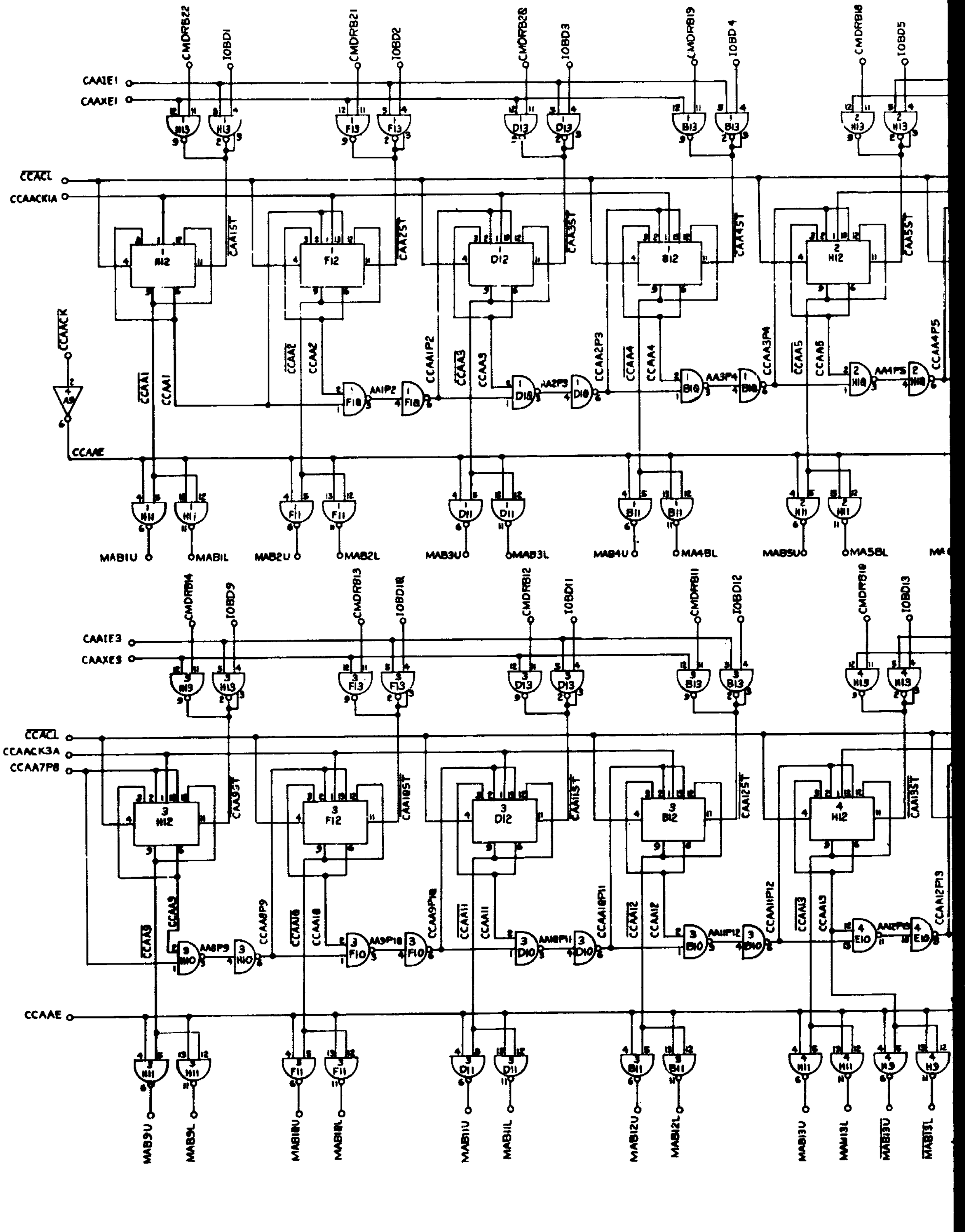


Figure A1—I/O subsystem block diagram.

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME

1



FOLDOUT FRAME 2

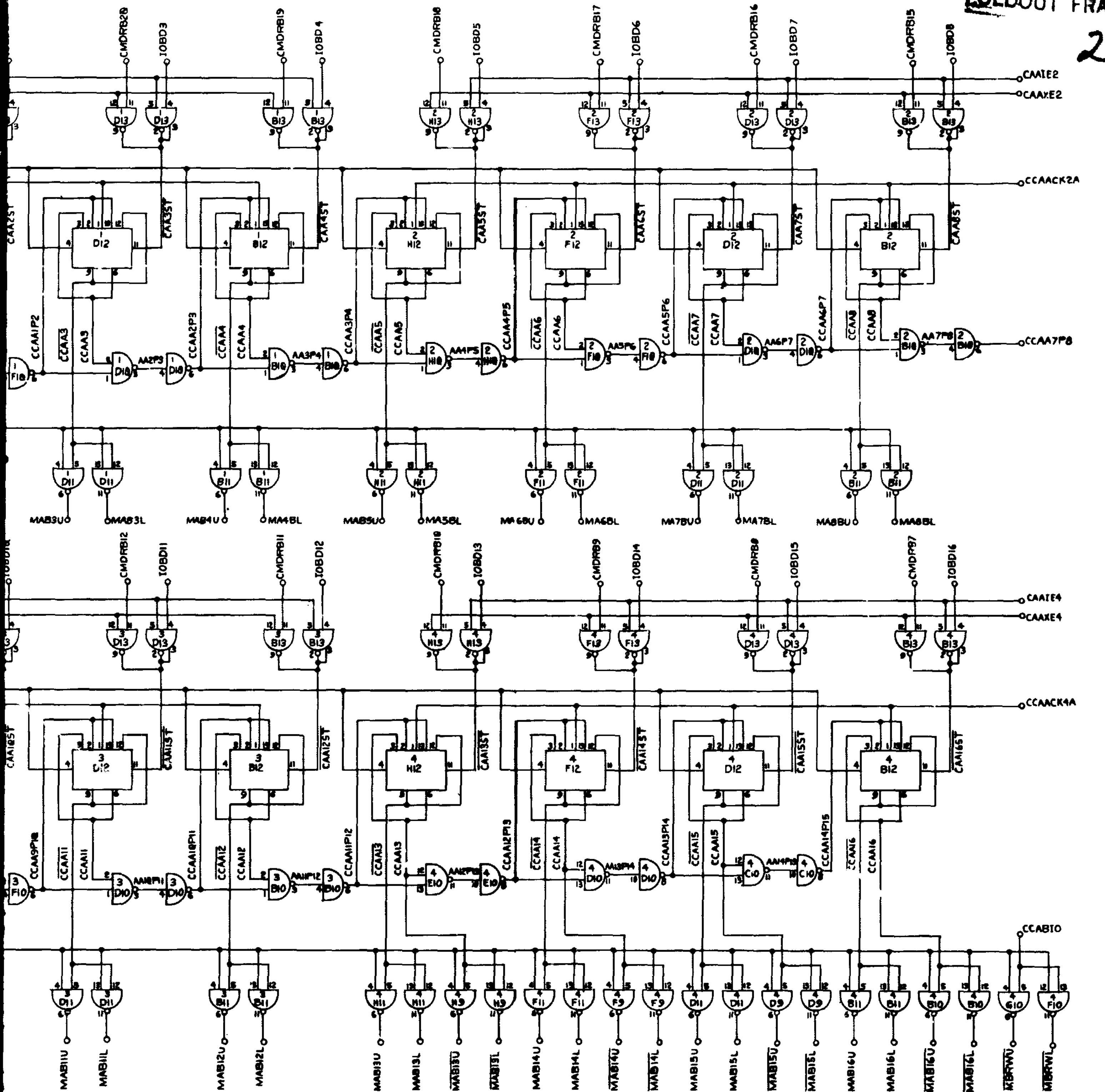
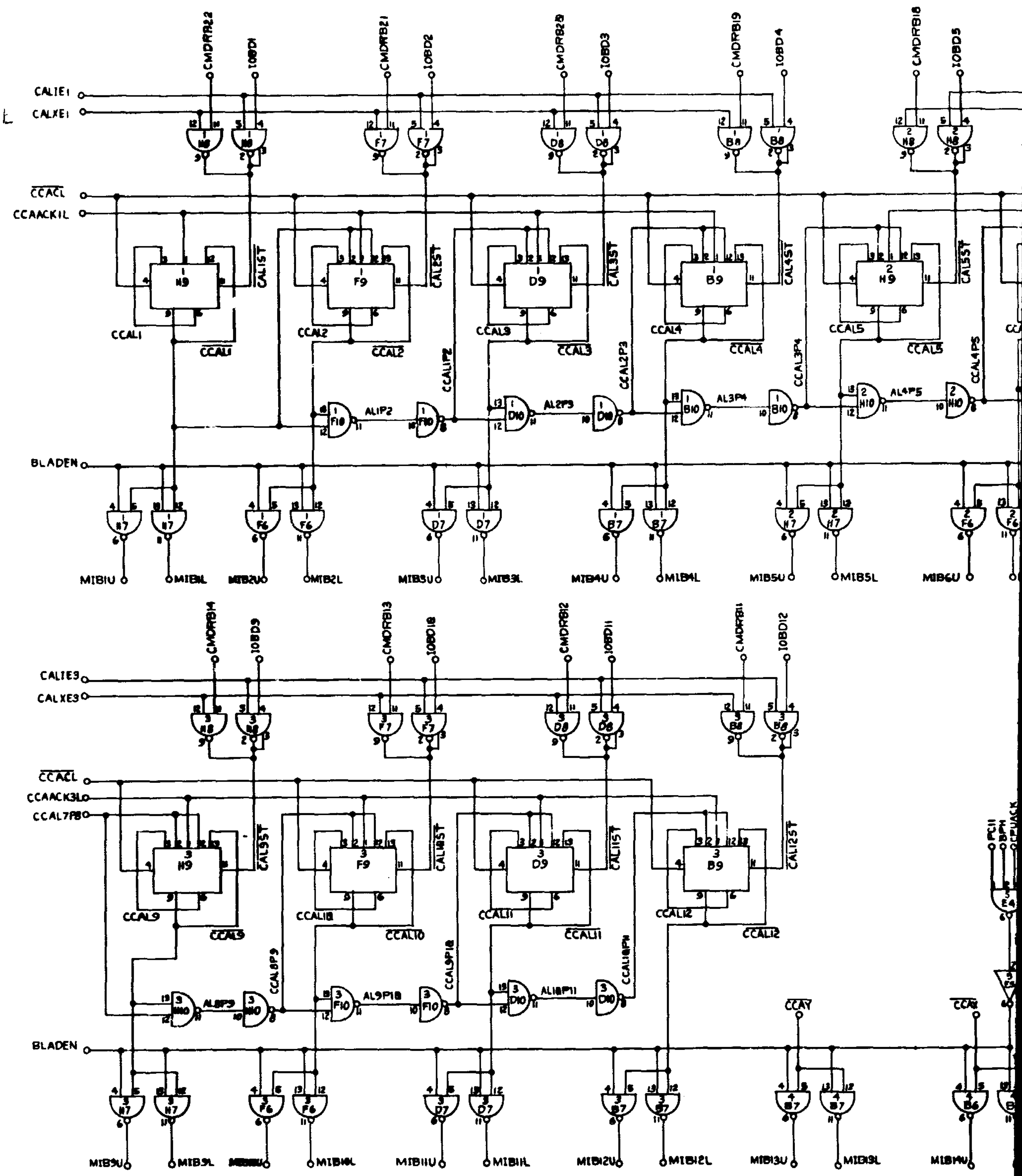


Figure A2 (GE1308724)–Logic diagram of CSCA address register.


CMDRB22
IOBD1
CMDRB21
IOBD2
CMDRB20
IOBD3
CMDRB19
IOBD4
CMDRB18
IOBD5
CALIE1
CALXE1
CCACL
CCAACK1L
CAL1ST
CAL2ST
CAL3ST
CAL4ST
CAL5ST
CCAL1
CCAL2
CCAL3
CCAL4
CCAL5
CCAL1P2
CCAL2P3
CCAL3P4
CCAL4P5
AL1P2
AL2P3
AL3P4
AL4P5
BLADEN
MIB1U
MIB1L
MIB2U
MIB2L
MIB3U
MIB3L
MIB4U
MIB4L
MIB5U
MIB5L
MIB6U
CMDRB14
IOBD9
CMDRB13
IOBD10
CMDRB12
IOBD11
CMDRB11
IOBD12
CALIE3
CALXE3
CCAACK3L
CCAL7P8
CAL9ST
CAL10ST
CAL11ST
CAL12ST
CCAL9
CCAL10
CCAL11
CCAL12
CCAL8P9
CCAL9P10
CCAL10P11
AL8P9
AL9P10
AL10P11
CCAY
MIB9U
MIB9L
MIB10L
MIB11U
MIB11L
MIB12U
MIB12L
MIB13U
MIB13L
MIB14U

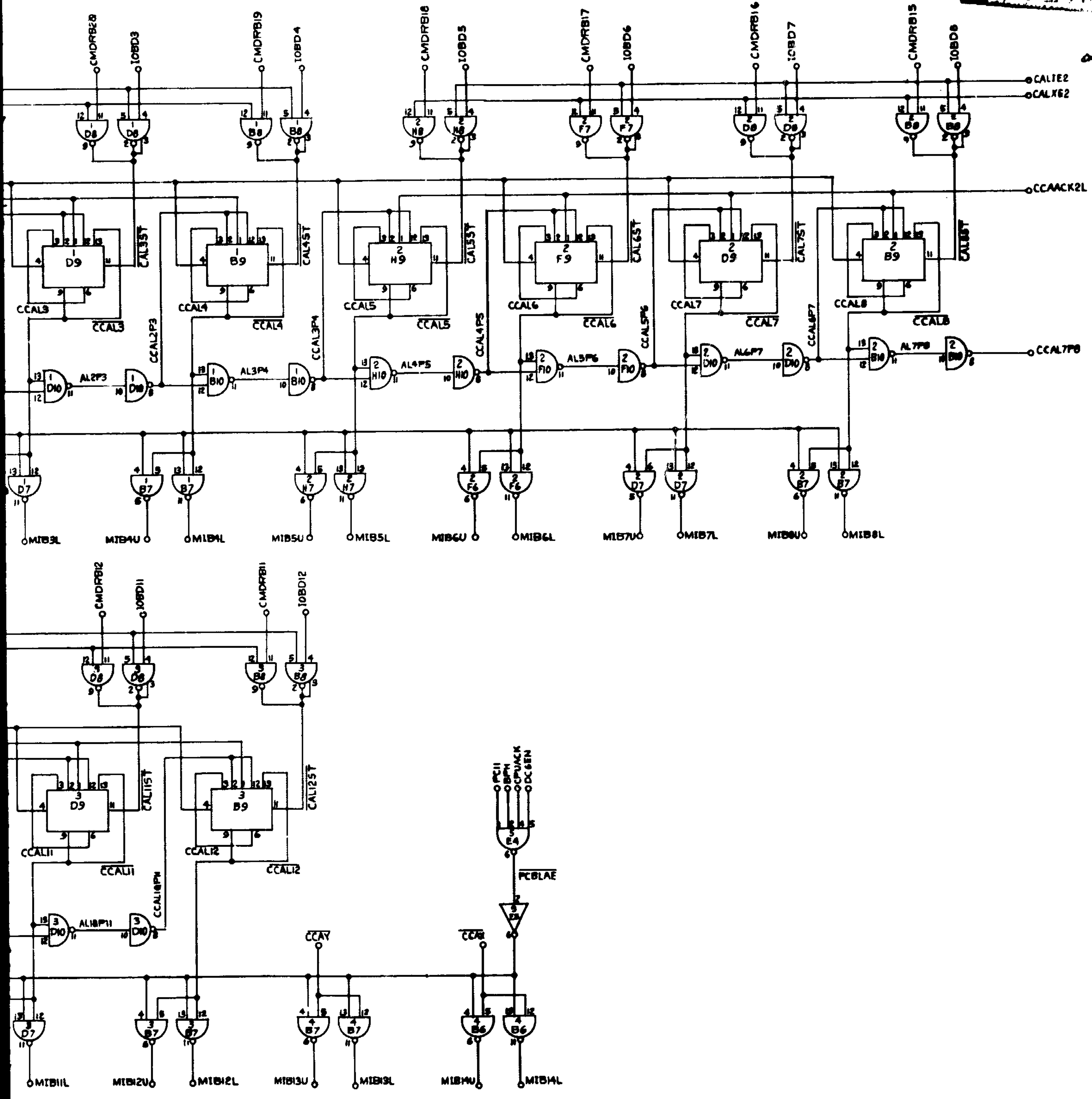
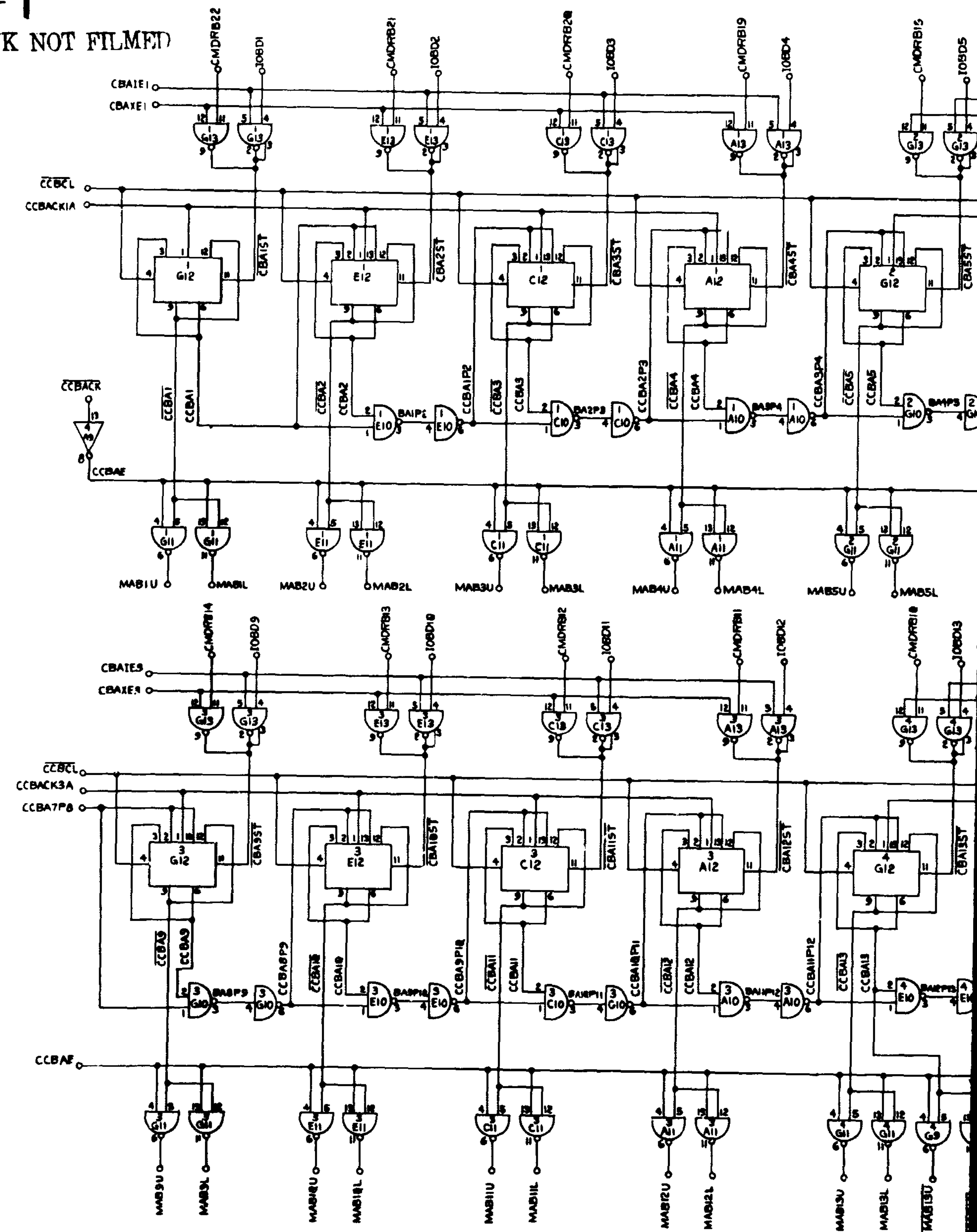


Figure A3 (GE1308725)—Logic diagram of CSCA block length register.

FOLDOUT FRAME 1

PRECEDING PAGE BLANK NOT FILMED



FOLDOUT FRAME 2

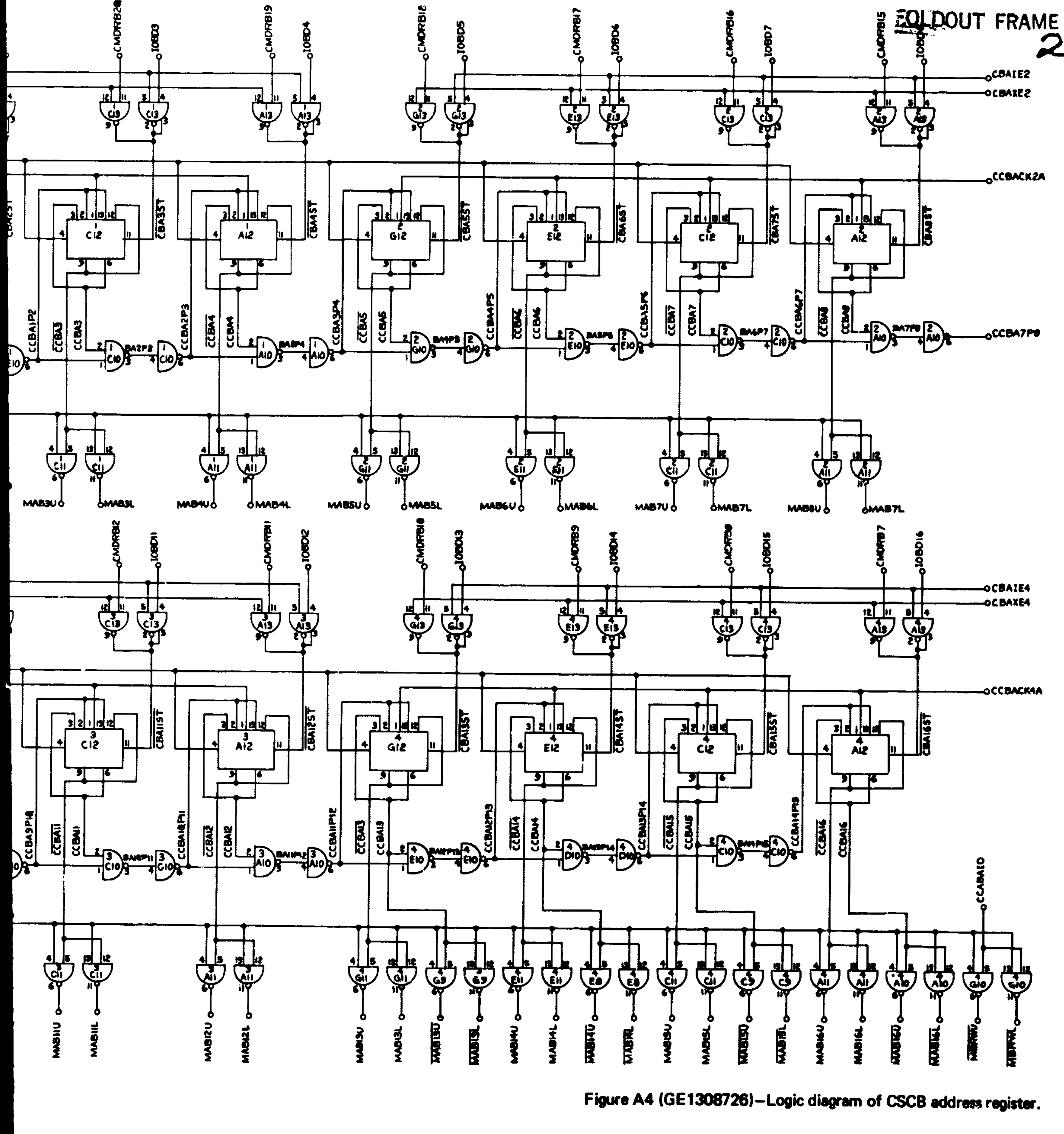


Figure A4 (GE1308726)–Logic diagram of CSCB address register.

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME

1

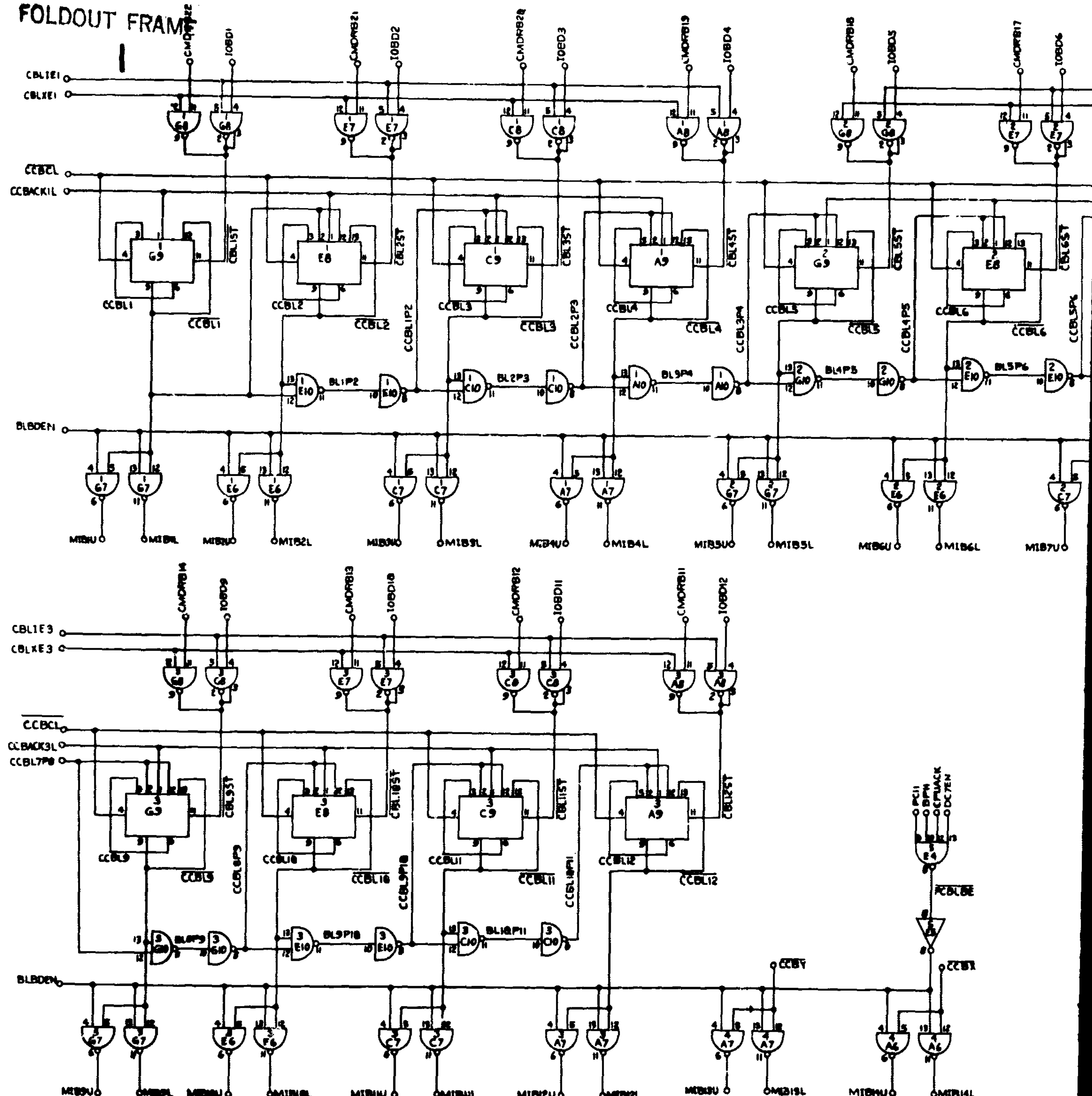


Figure A5 (G

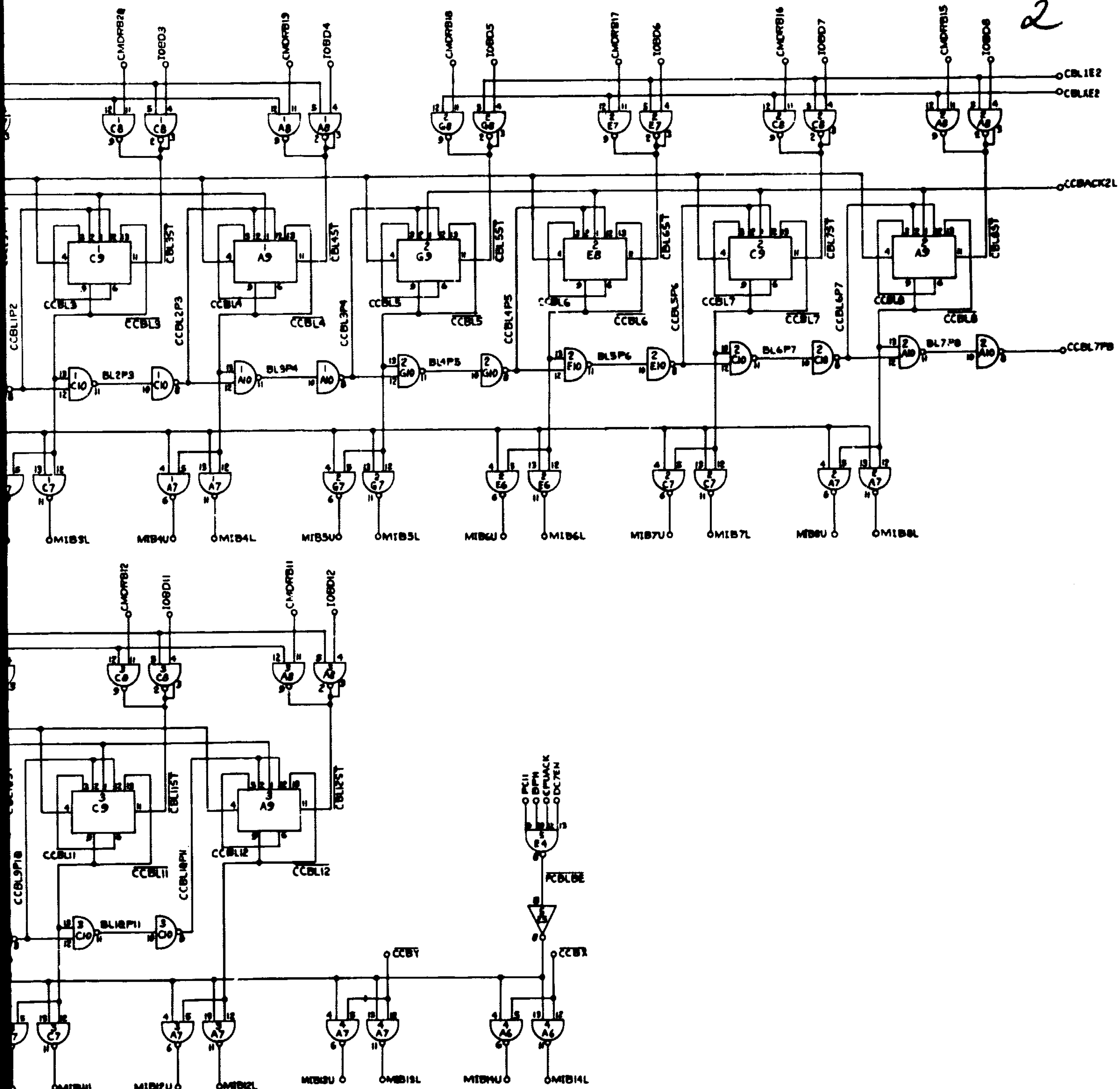


Figure A5 (GE1308727)—Logic diagram of CSCB block length register.

PRECEDING PAGE BLANK NOT FILMED

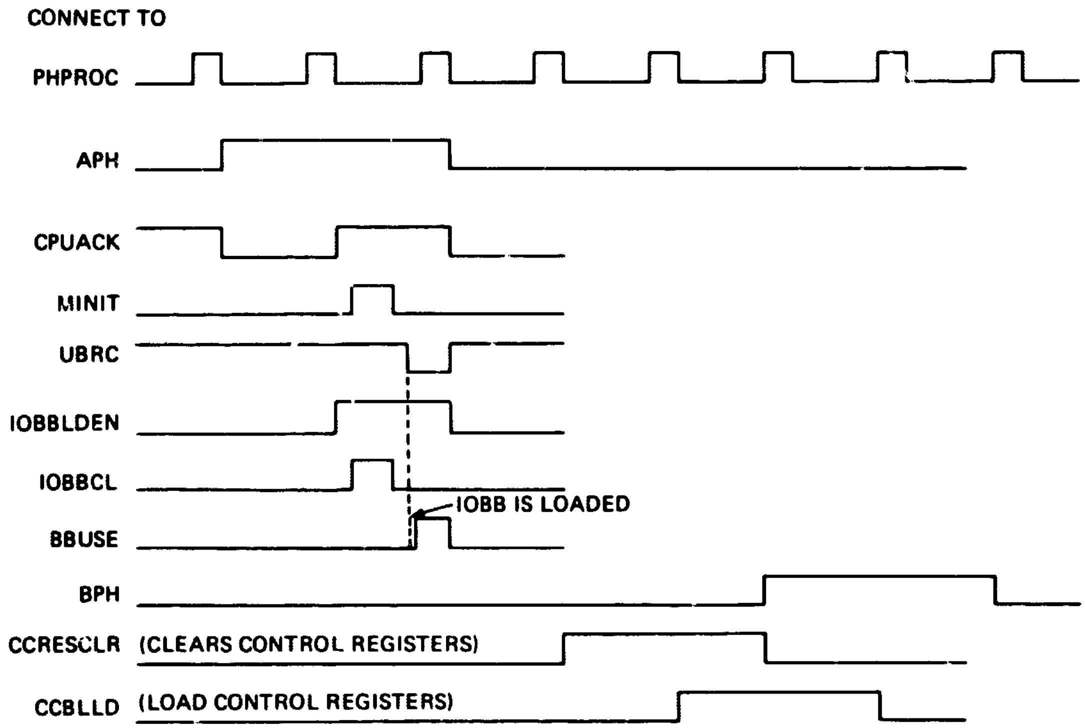
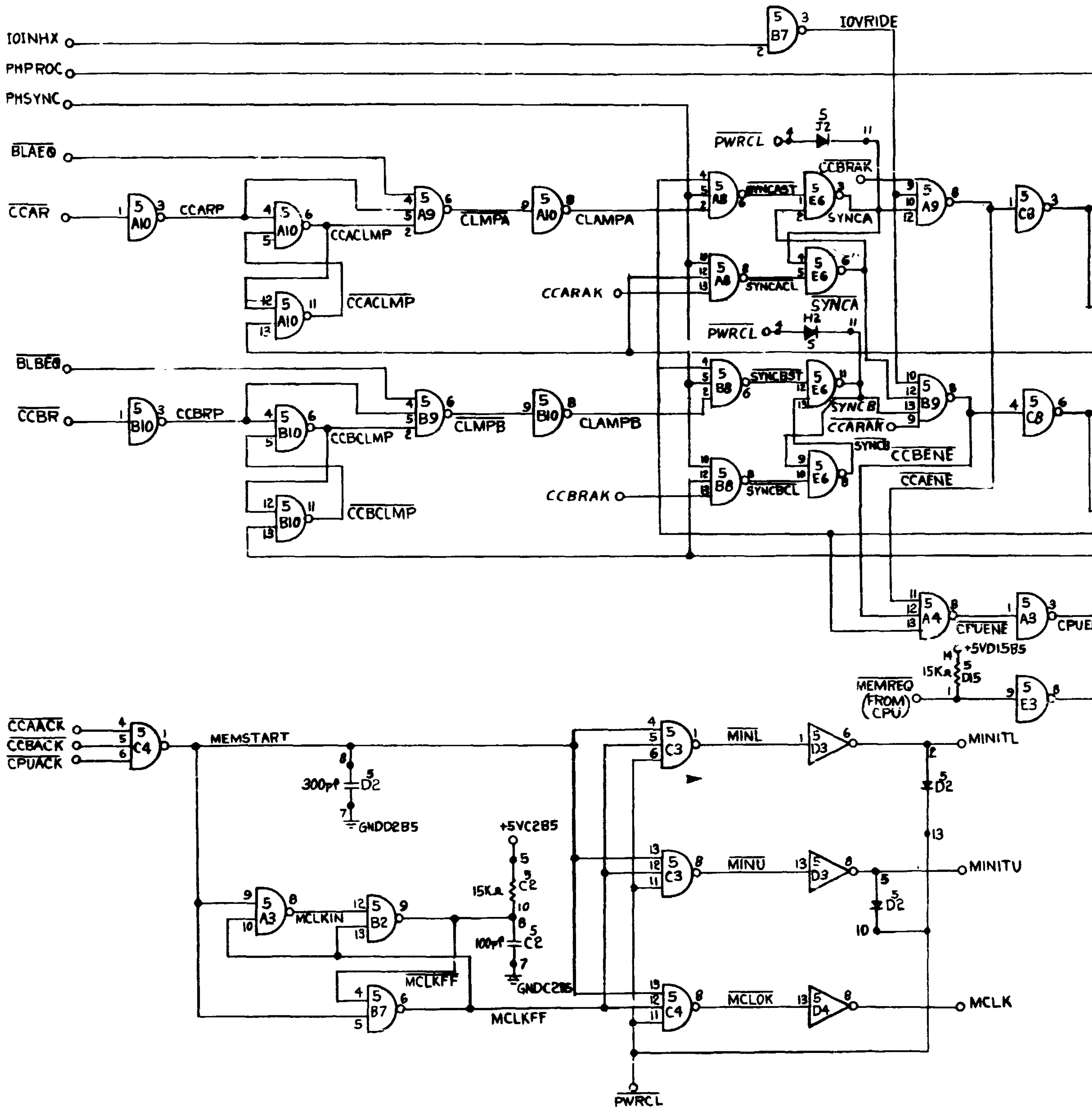


Figure A6—Timing diagram.

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME 1



FOLDOUT FRAME 2

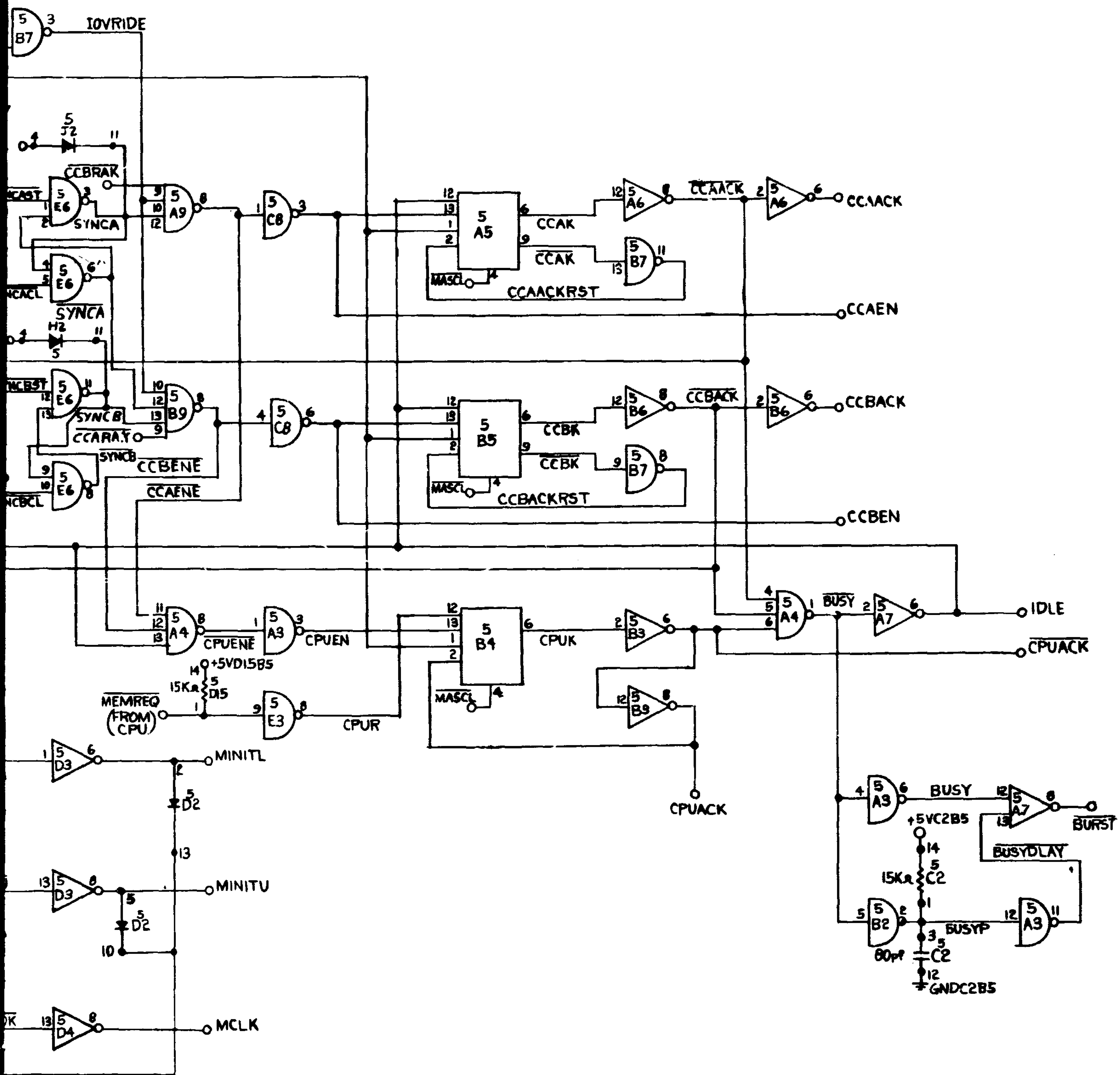


Figure A7 (GD1136184)—Bus control logic diagram.

FOLDOUT FRAME

1

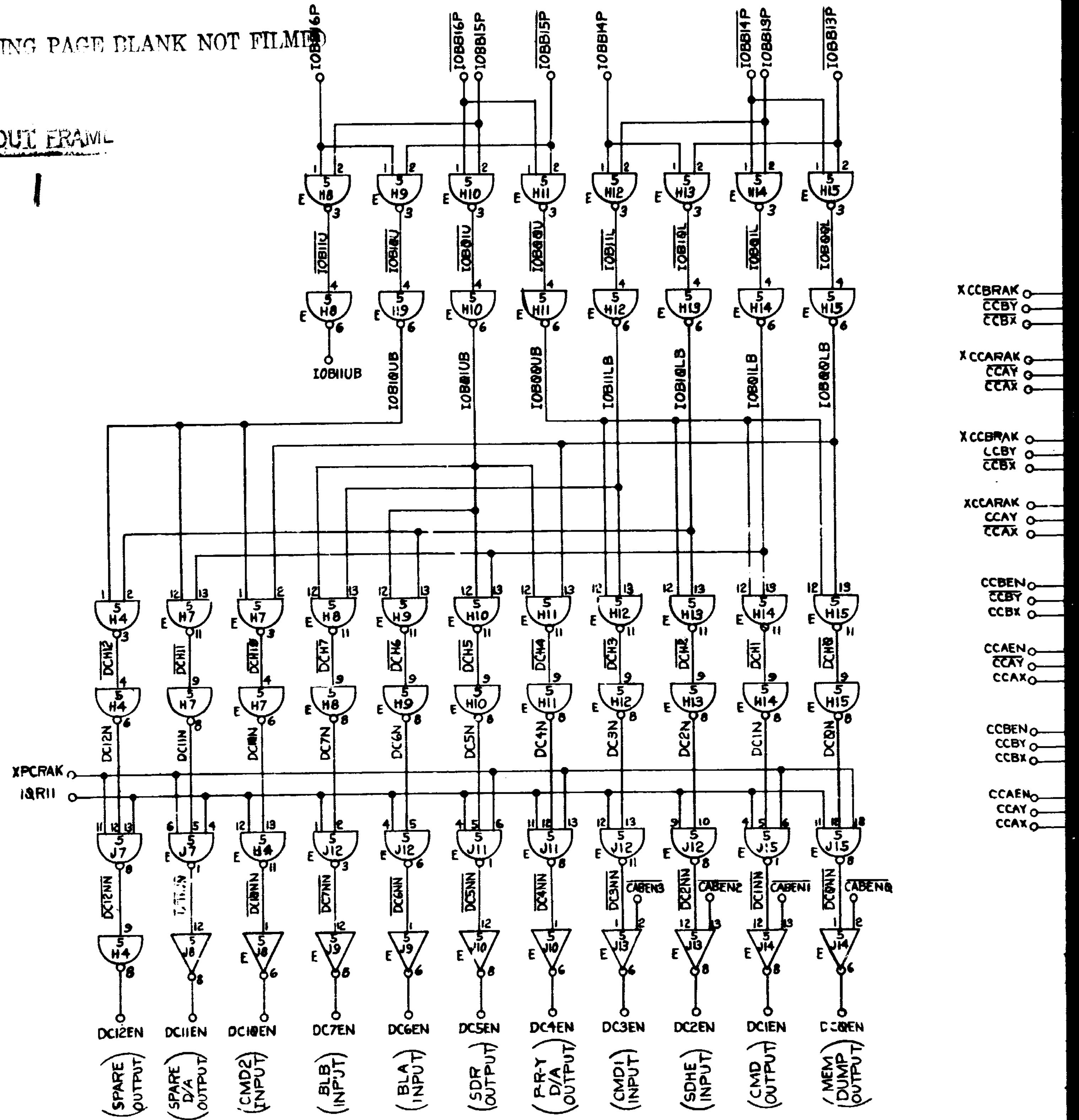


Figure A8 (GD1136182)—Logic diagram c

OT FILMED

FOLDOUT FRAME

2

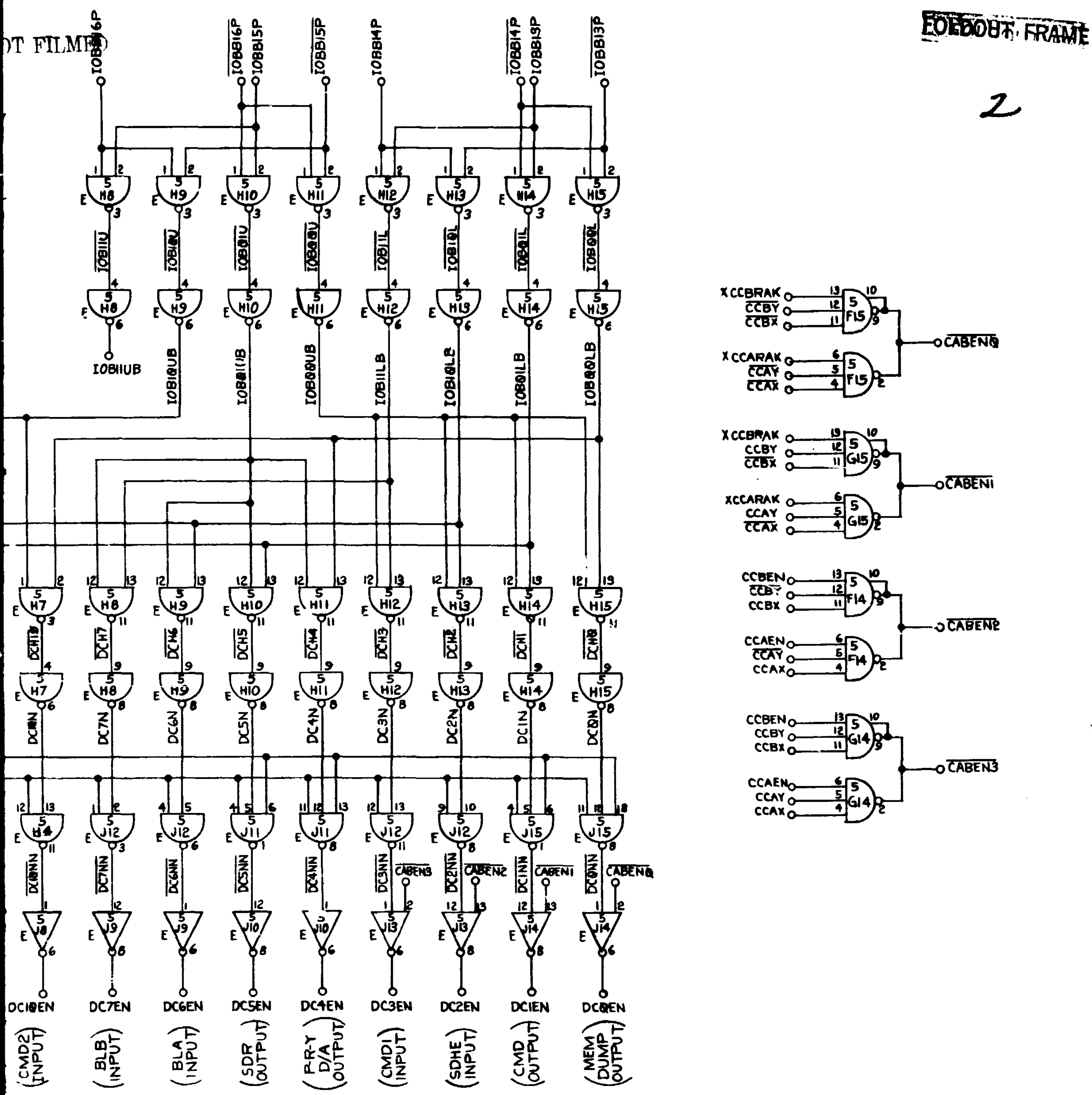


Figure A8 (GD1136182)—Logic diagram of control channels enable.

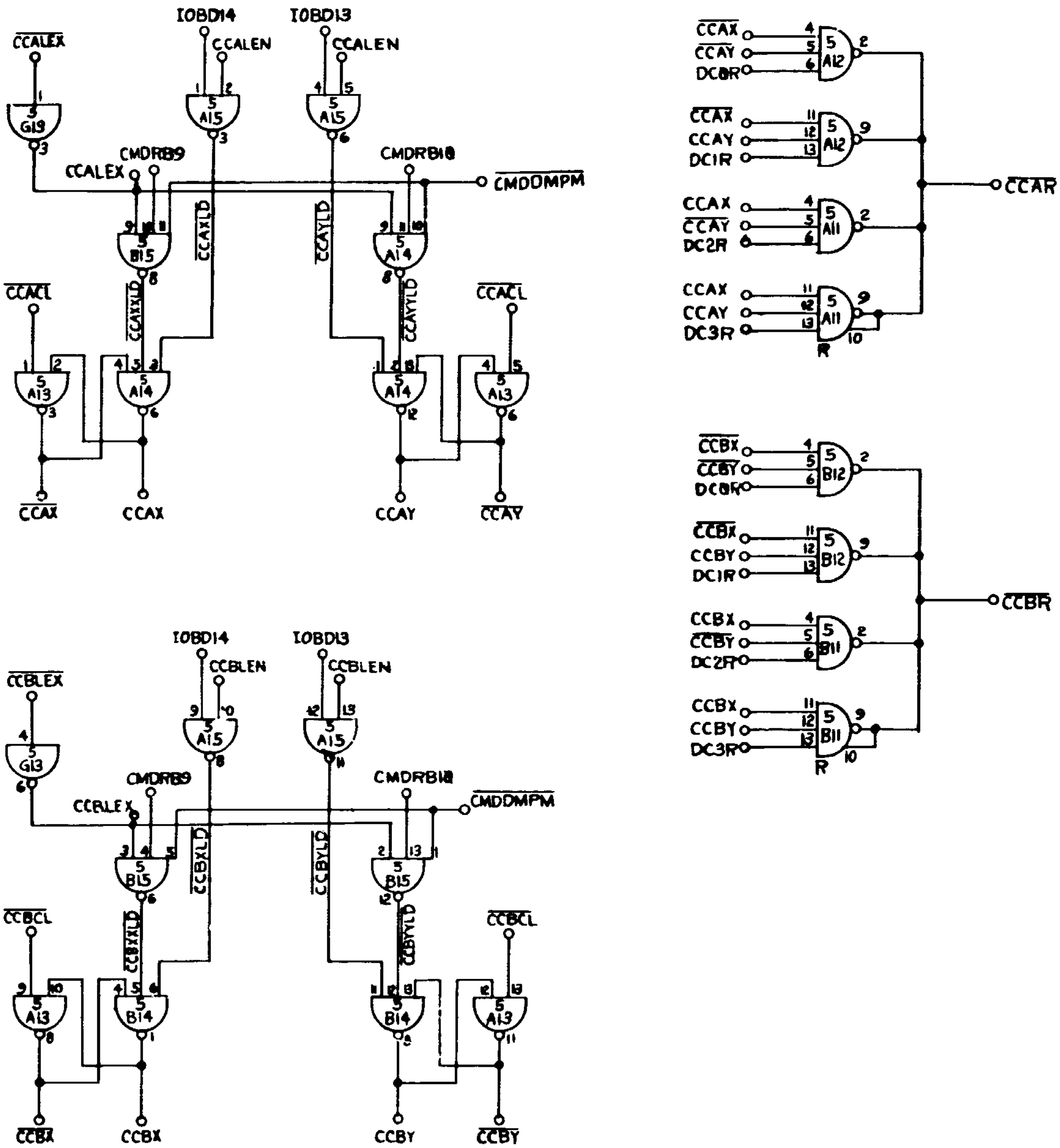


Figure A9 (GD1136185)—Logic diagram of control channels cycle request and device select.

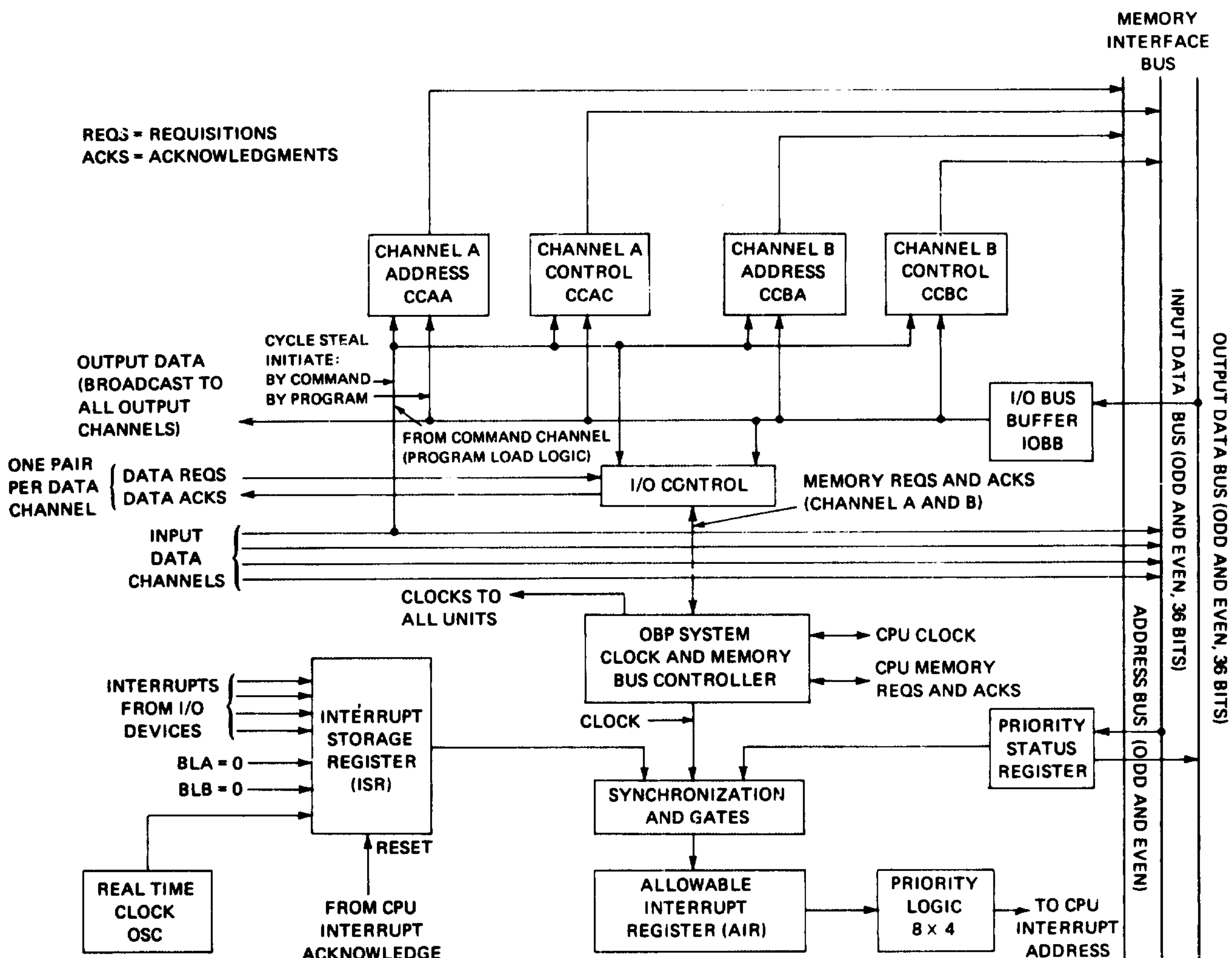
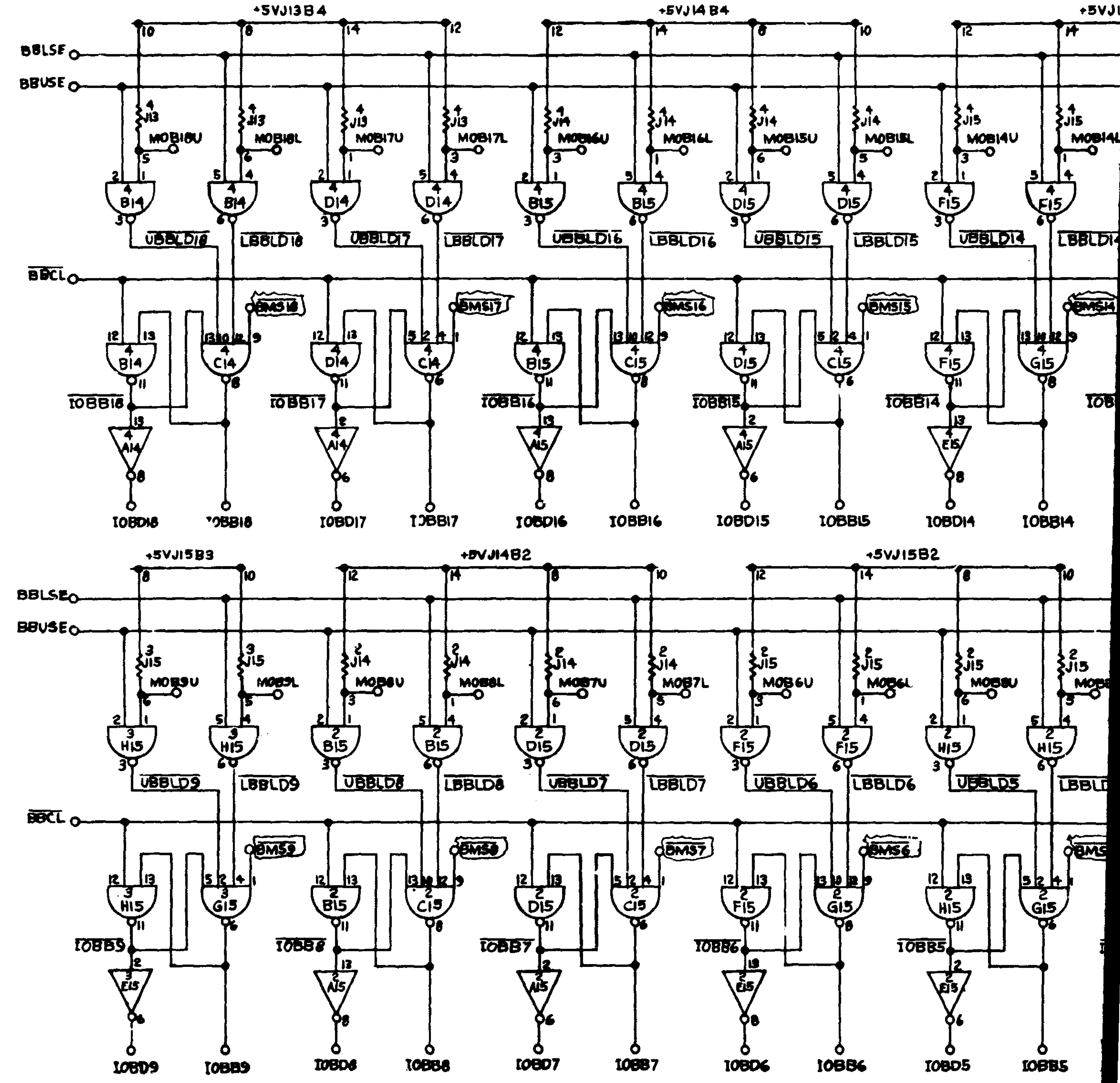


Figure A10—I/O unit functional block diagram.

FOLDOUT FRAME 1

[redacted]



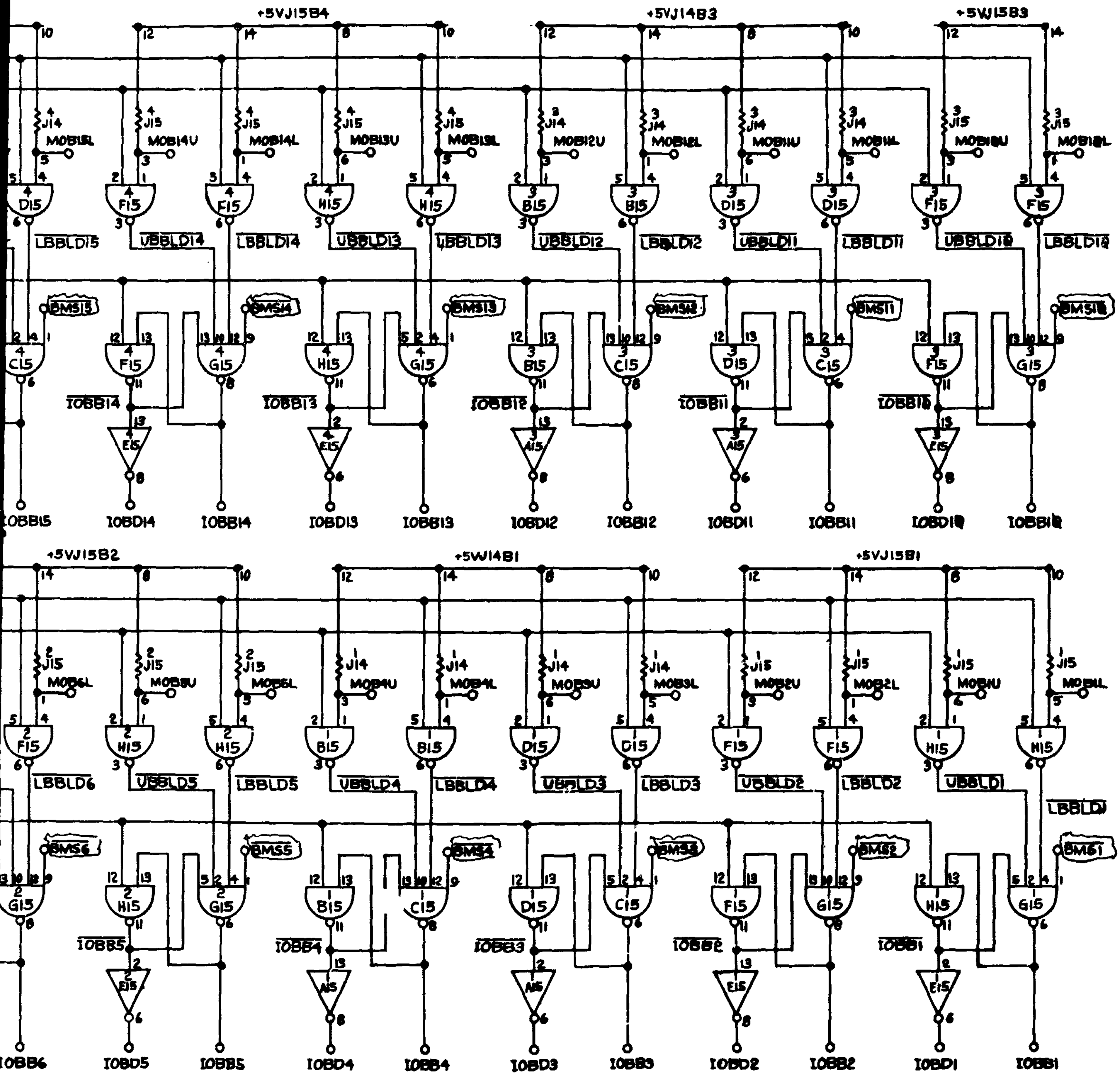
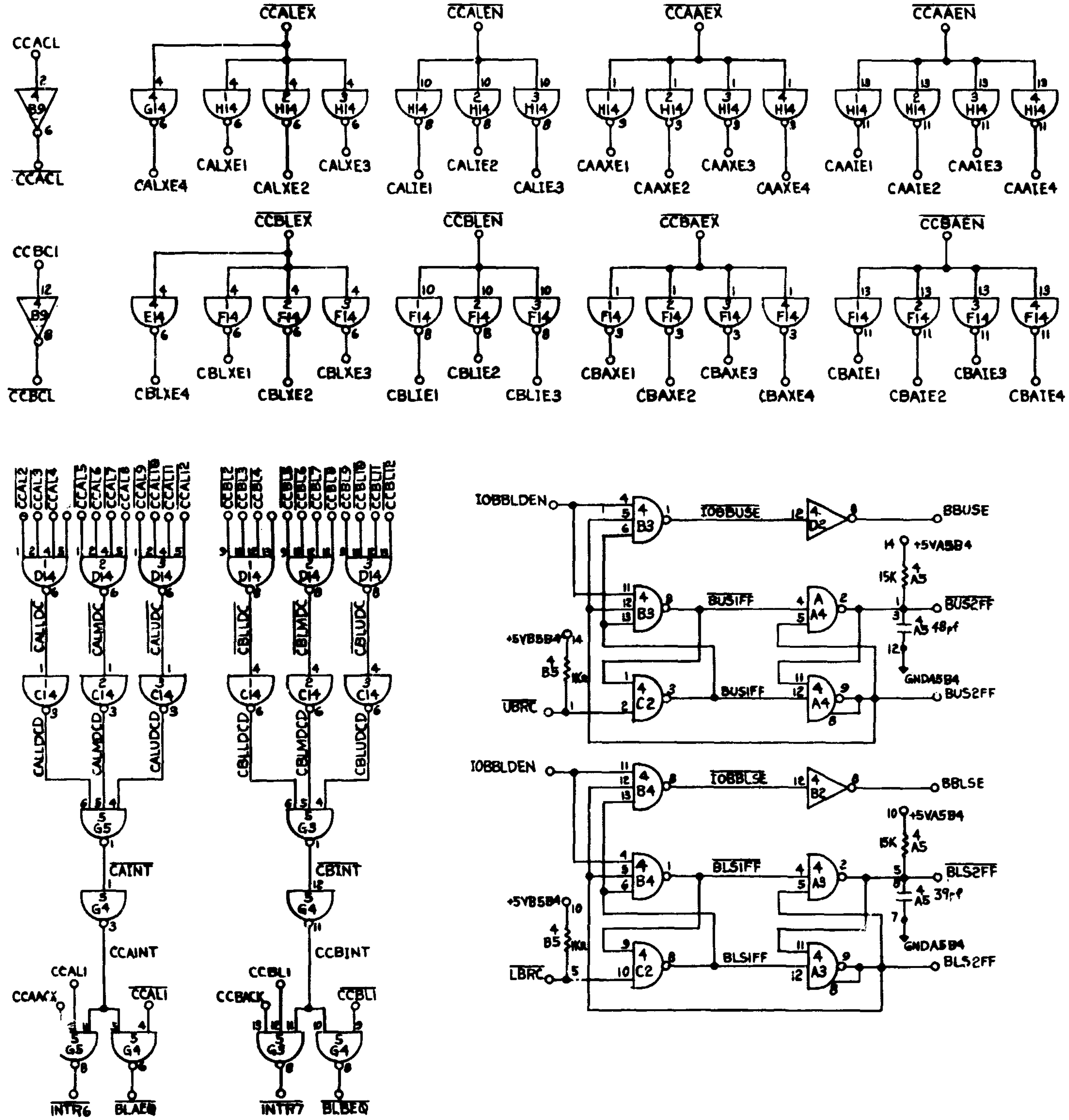


Figure A11 (GD1308695)—Logic diagram of I/O bus buffer.

PRECEDING PAGE BLANK NOT FILMED

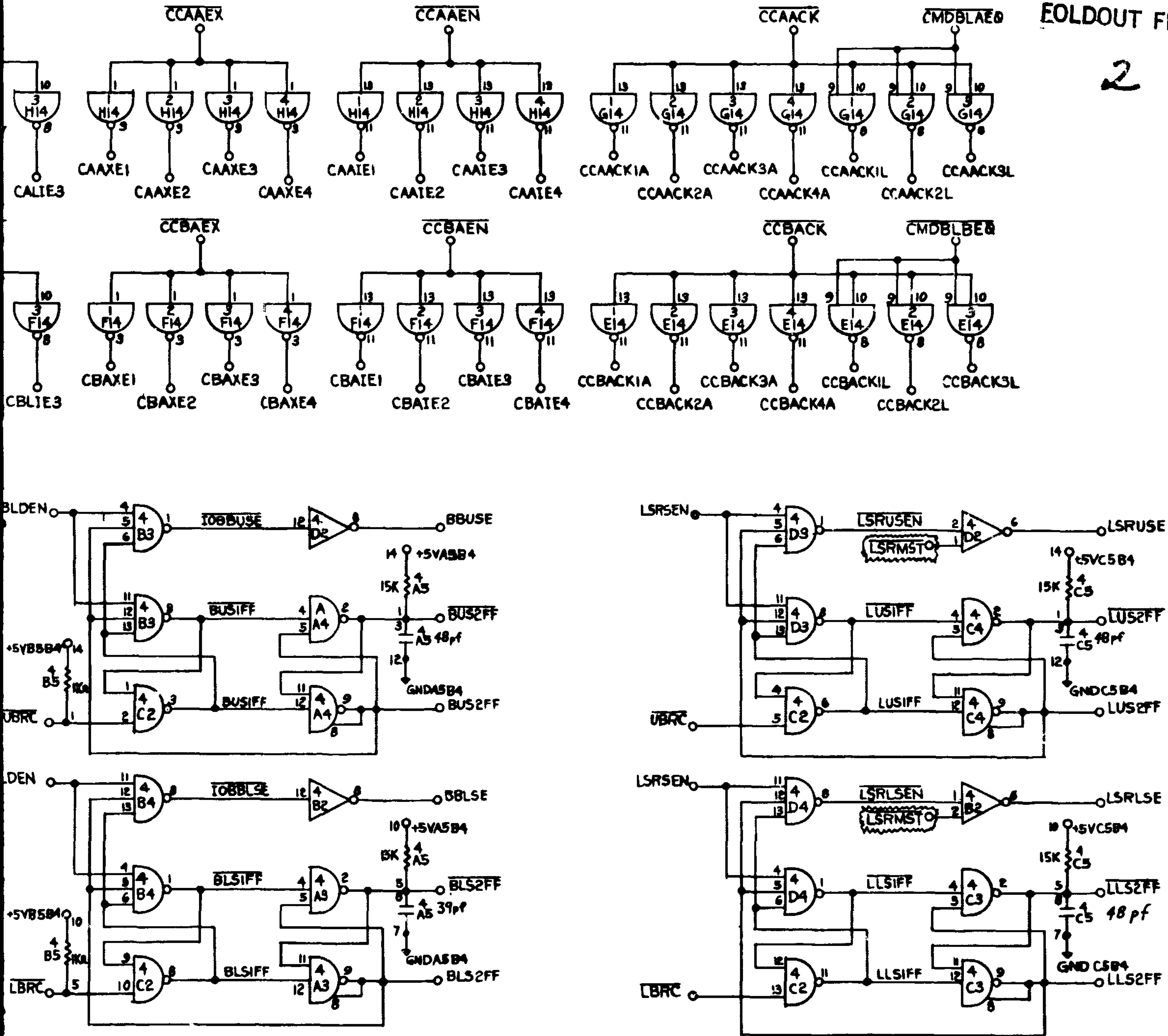
FOLDOUT FRAME 1



NOTE: UNLESS SPECIFIED ALL RESISTORS ARE 15KΩ

FOLDOUT FRAME

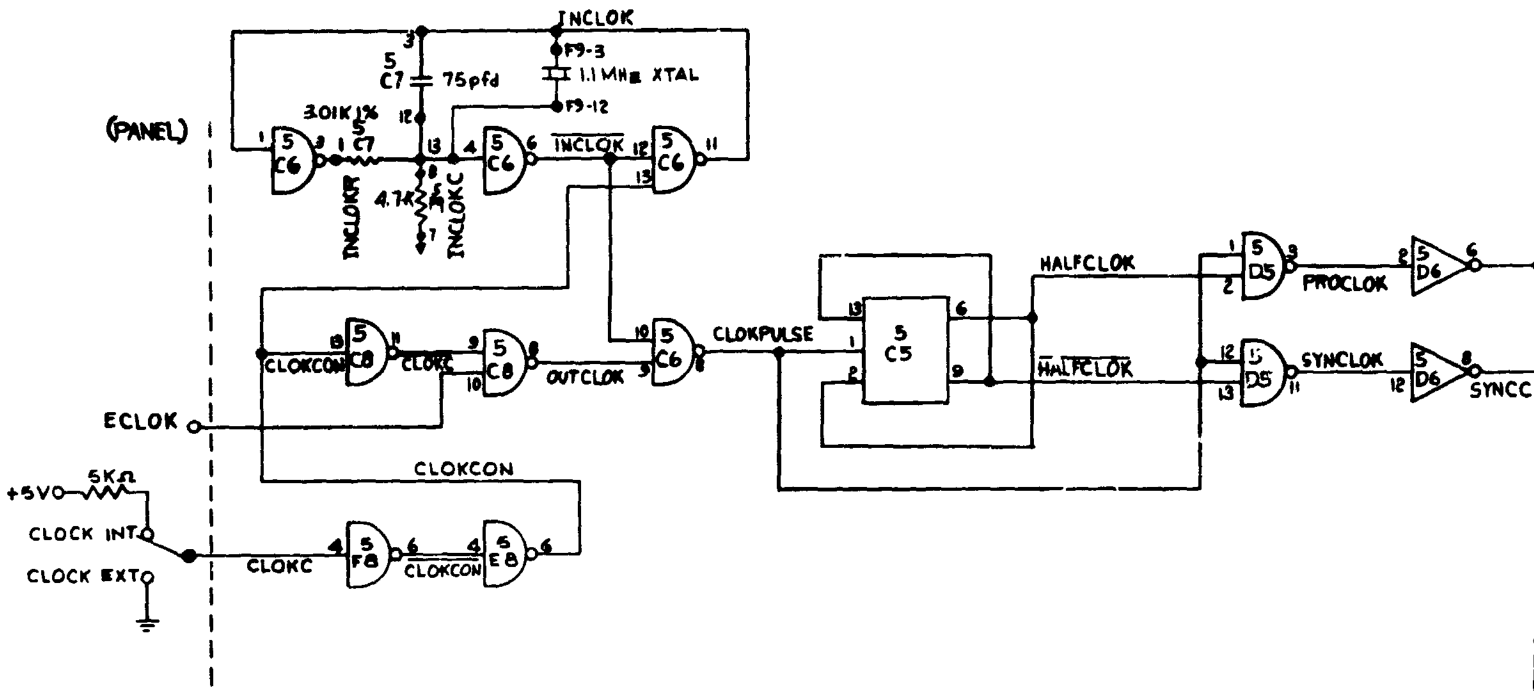
2



ure A12 (GD1308576)—Control logic for CC, LSR, and IOBB registers.

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME 1



FOLDOUT FRAME 2

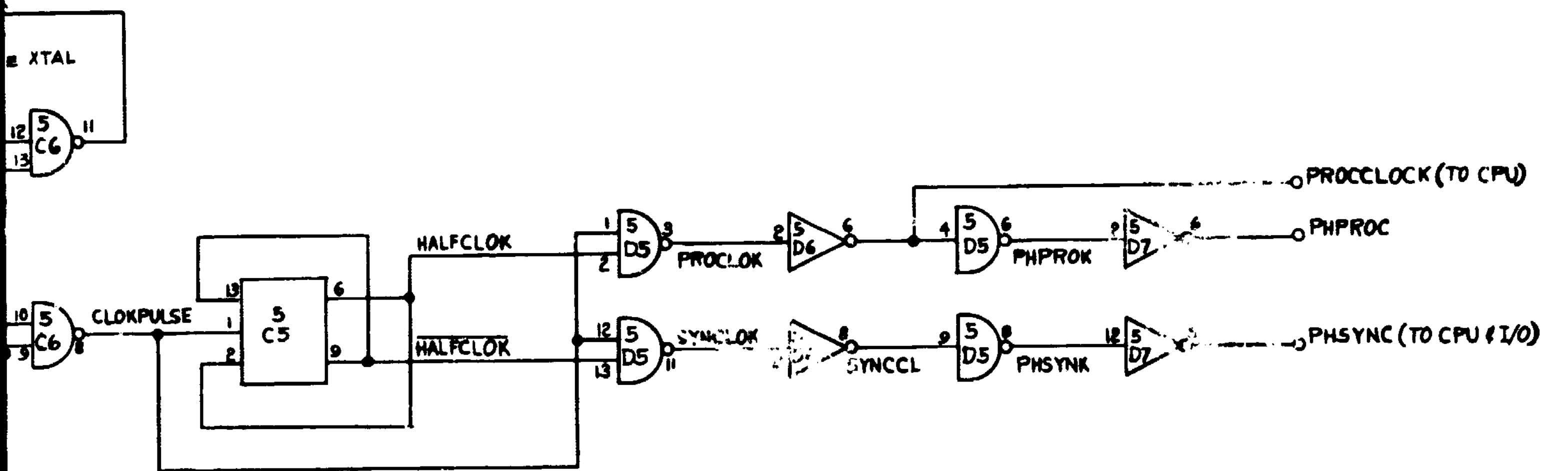
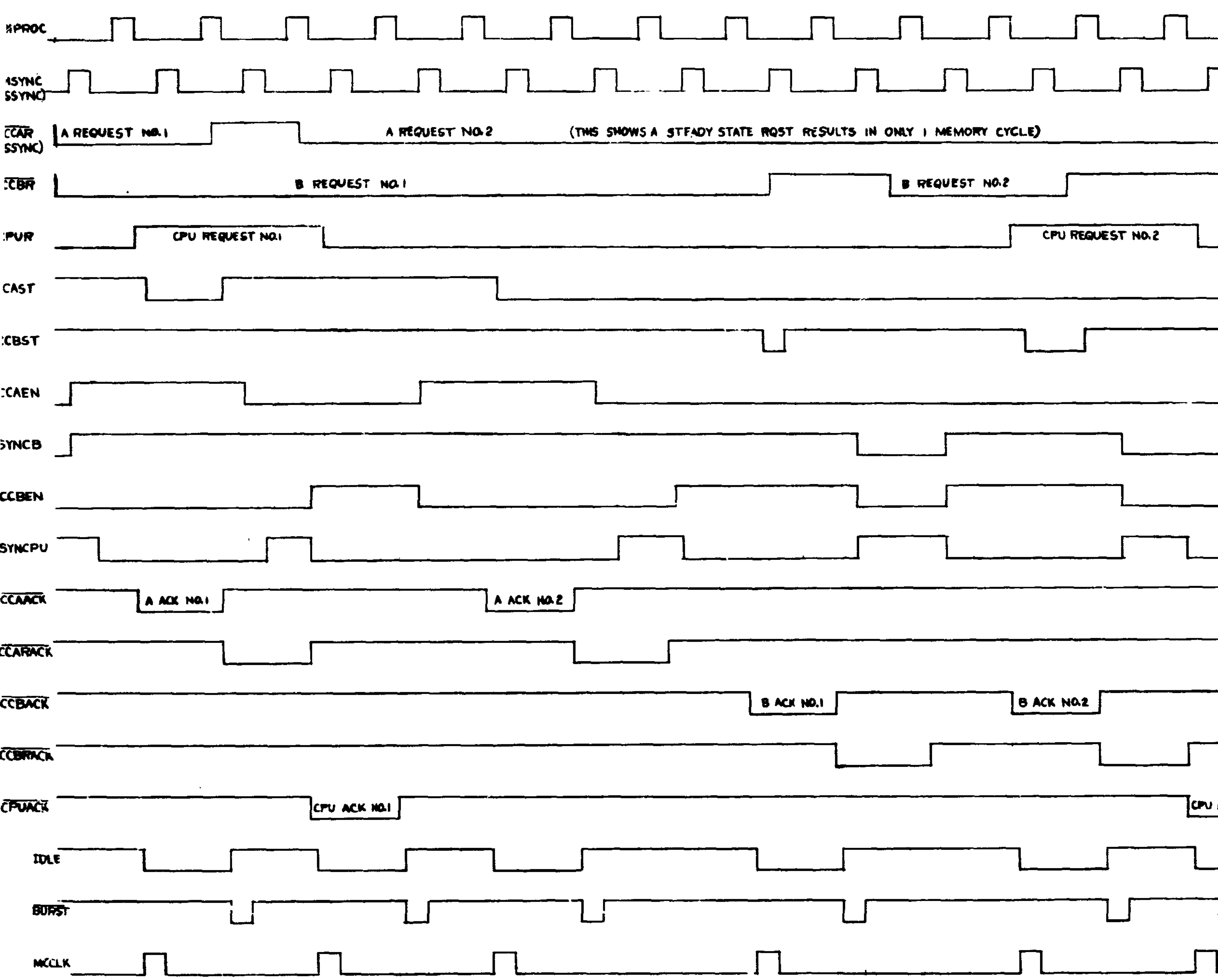


Figure A13 (GD11?6183)–System clock logic diagram.

PRECEDING PAGE BLANK NOT FILMED

WAVEFORMS:1A° = 1.0μSEC FOR 1.00 MHz CLOCK



OPERATIONAL CHARACTERISTICS FOR CHANNEL A&B CONTROL SIGNALS

THE REQUESTS MUST DROP AND REMAIN LOW UNTIL OCCURANCE OF THE ACKNOWLEDGE PULSE. FOR ADDITIONAL REQUEST MUST GO TO +5VOLTS FOR A MINIMUM OF 2.00μSEC BEFORE DROPPING; OTHERWISE THE REQUEST MAY NOT BE ACKNOWLEDGED. RISE AND FALL TIMES ARE NOT CRITICAL BUT CANNOT BE INCLUDED IN THE 2.00μSEC

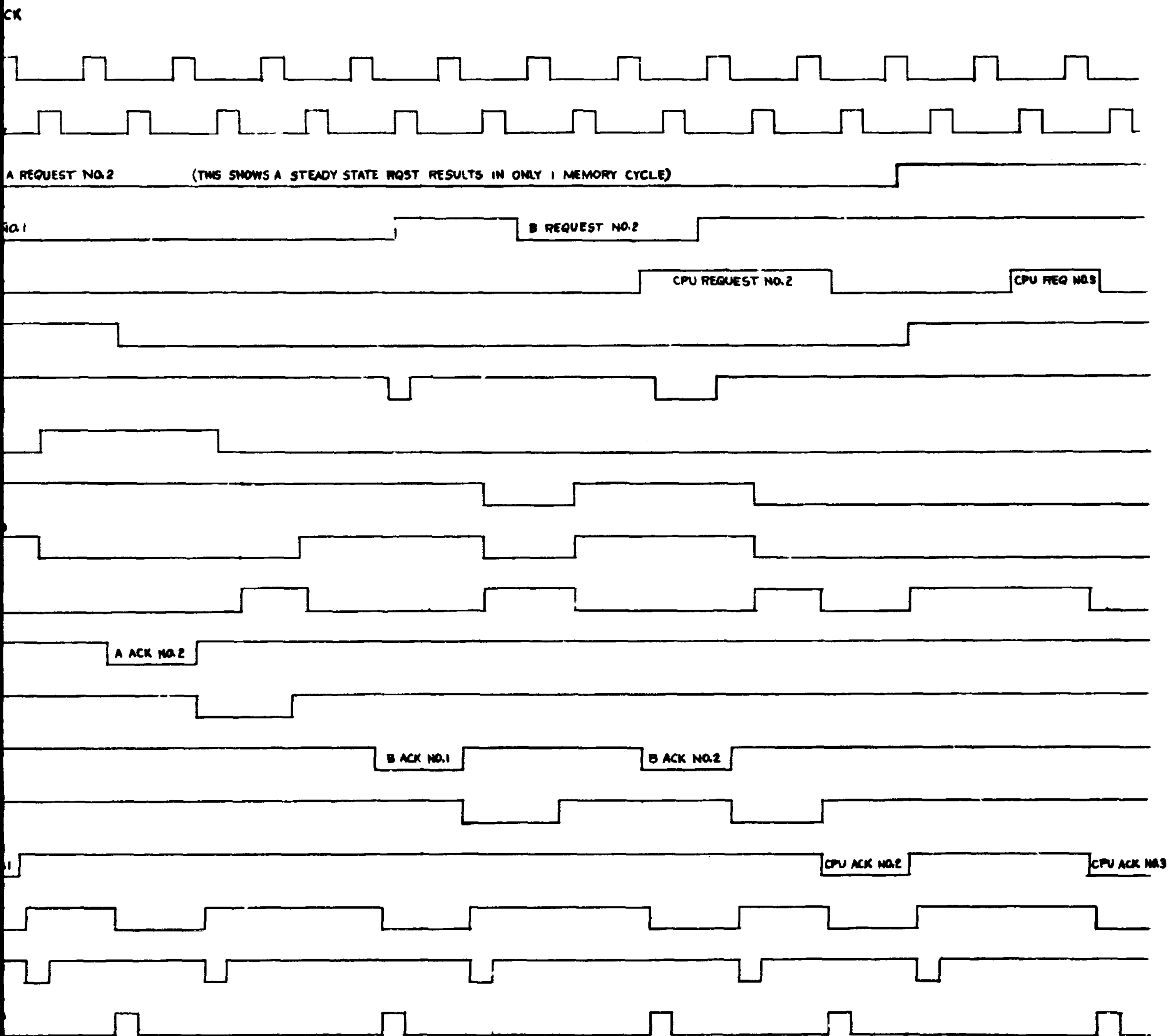
OPERATIONAL CHARACTERISTICS FOR CPU CONTROL SIGNAL

THE WAVEFORM DEPICTS THE LEADING AND TRAILING EDGE TIMING OF TYPICAL REQUESTS. A STEADY STATE +5V HOWEVER, RESULTS IN CONTINOUS CPU ACKNOWLEDGES

* THESE REQUESTS MAY BE ASSYNCHRONOUS

Figure A14 (GD1136189)–

FILMED



TROL SIGNALS

OCCURANCE OF
ST MUST GO TO
ROPPING;
EDGED. RISE
E INCLUDED IN

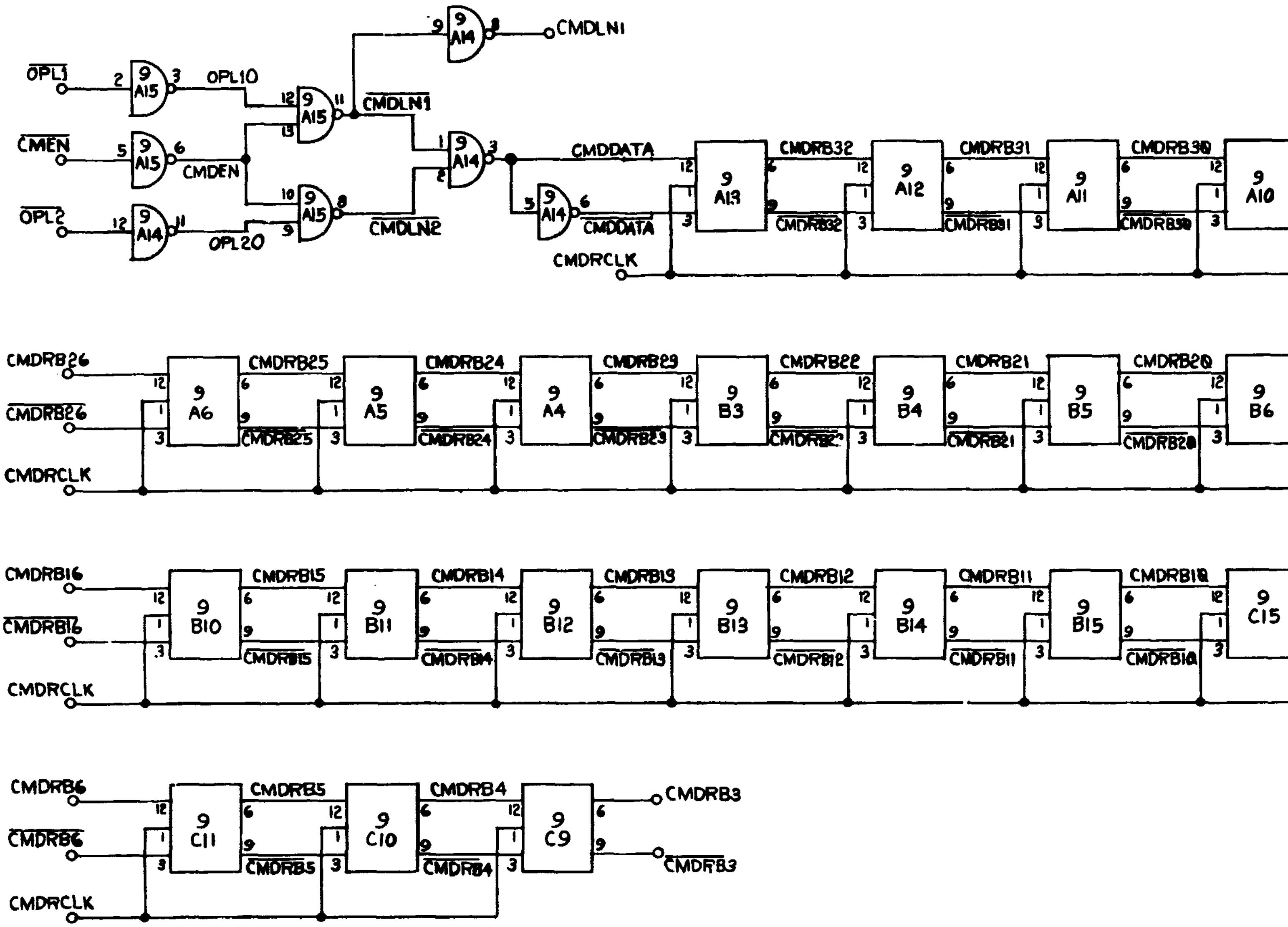
OPERATIONAL CHARACTERISTICS FOR CPU CONTROL SIGNAL

THE WAVEFORM DEPICTS THE LEADING AND TRAILING EDGE TIMING OF TYPICAL REQUESTS. A STEADY STATE +5V HOWEVER, RESULTS IN CONTINOUS CPU ACKNOWLEDGES

Figure A14 (GD1136189)—Clock and bus controller timing diagram.

FOLDOUT FRAME 1

PRECEDING PAGE BLANK NOT FILMED



FOLDOUT FRAME 2

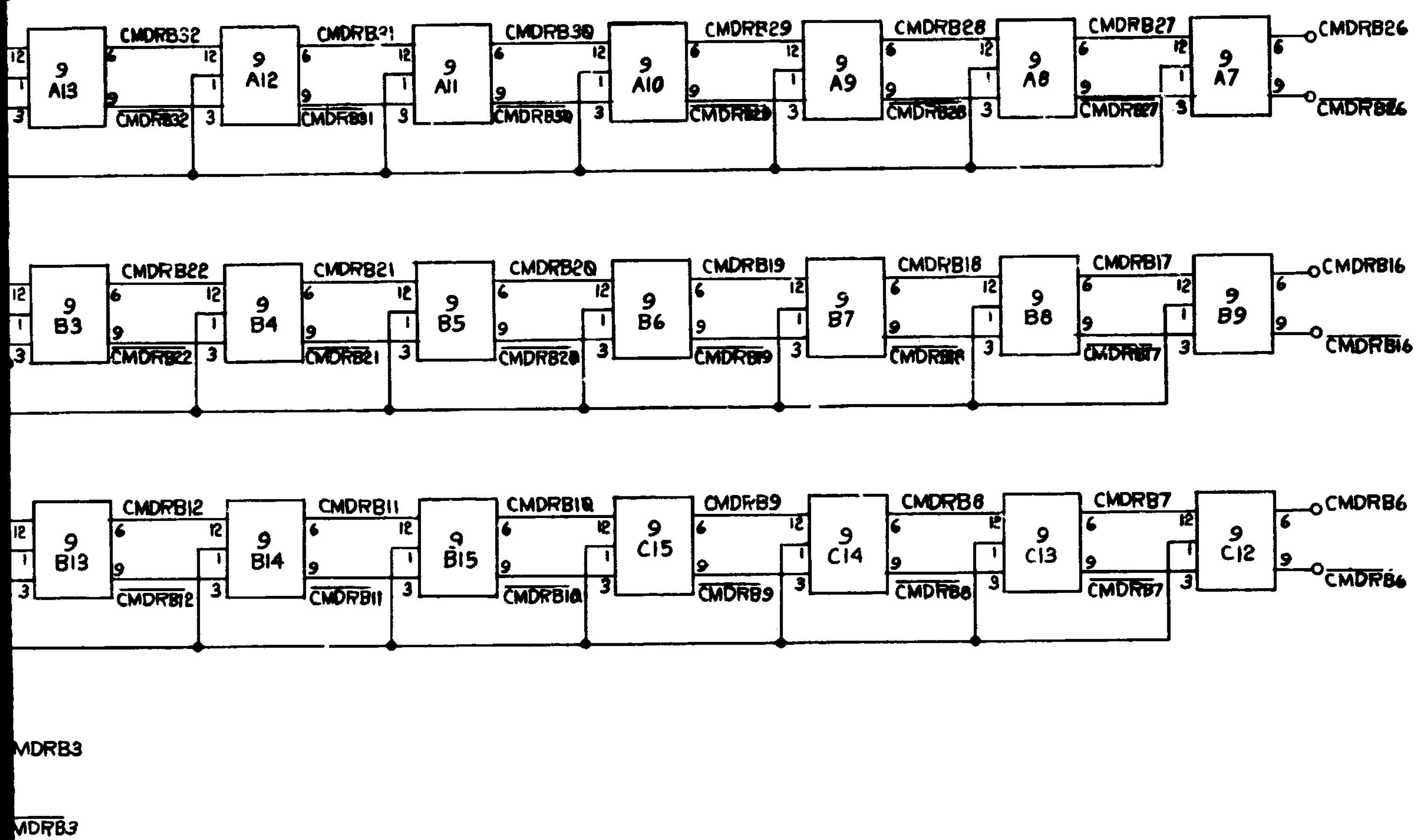
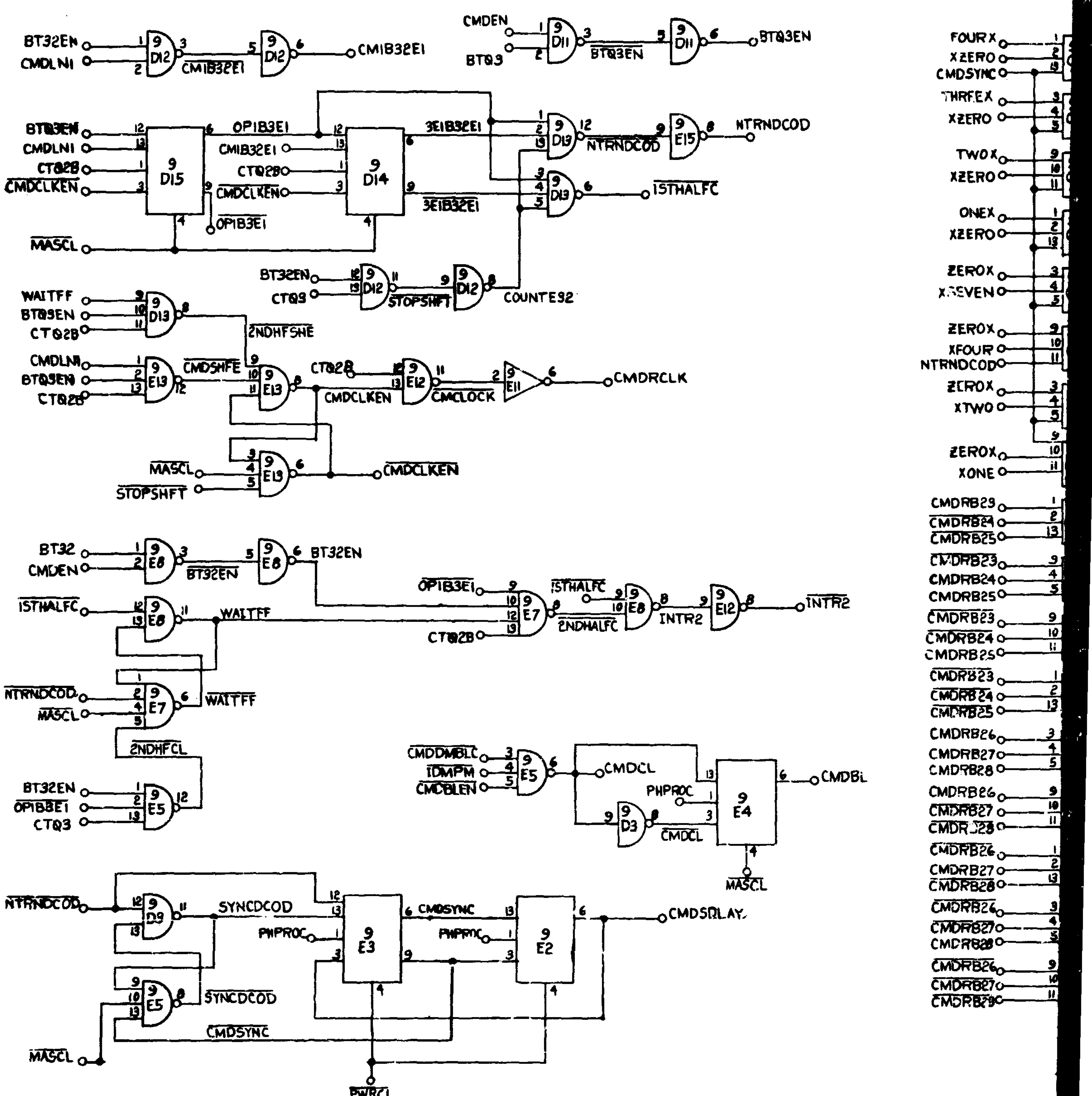


Figure A15 (GD1308597)—Command register logic diagram.

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME 1



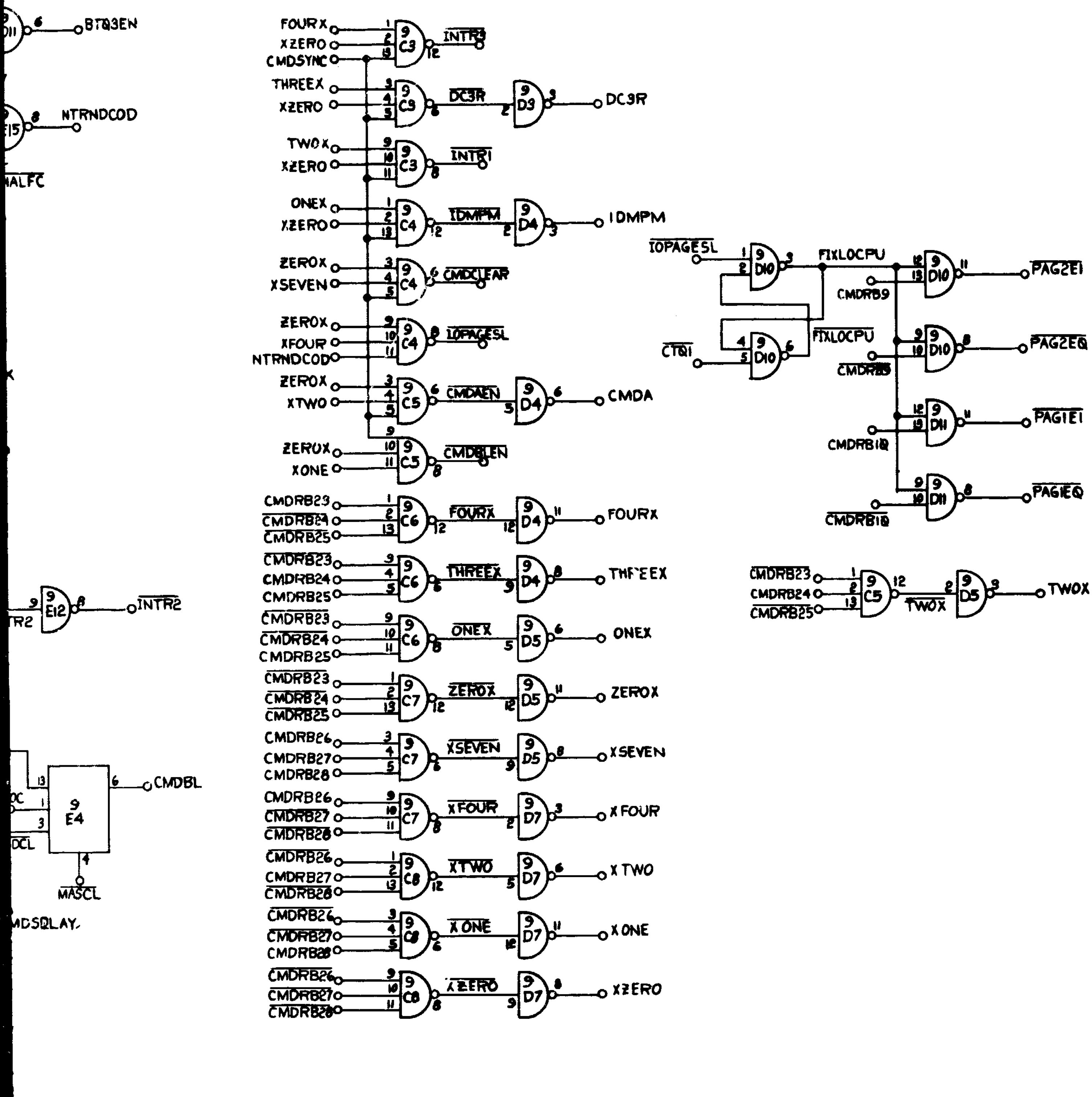
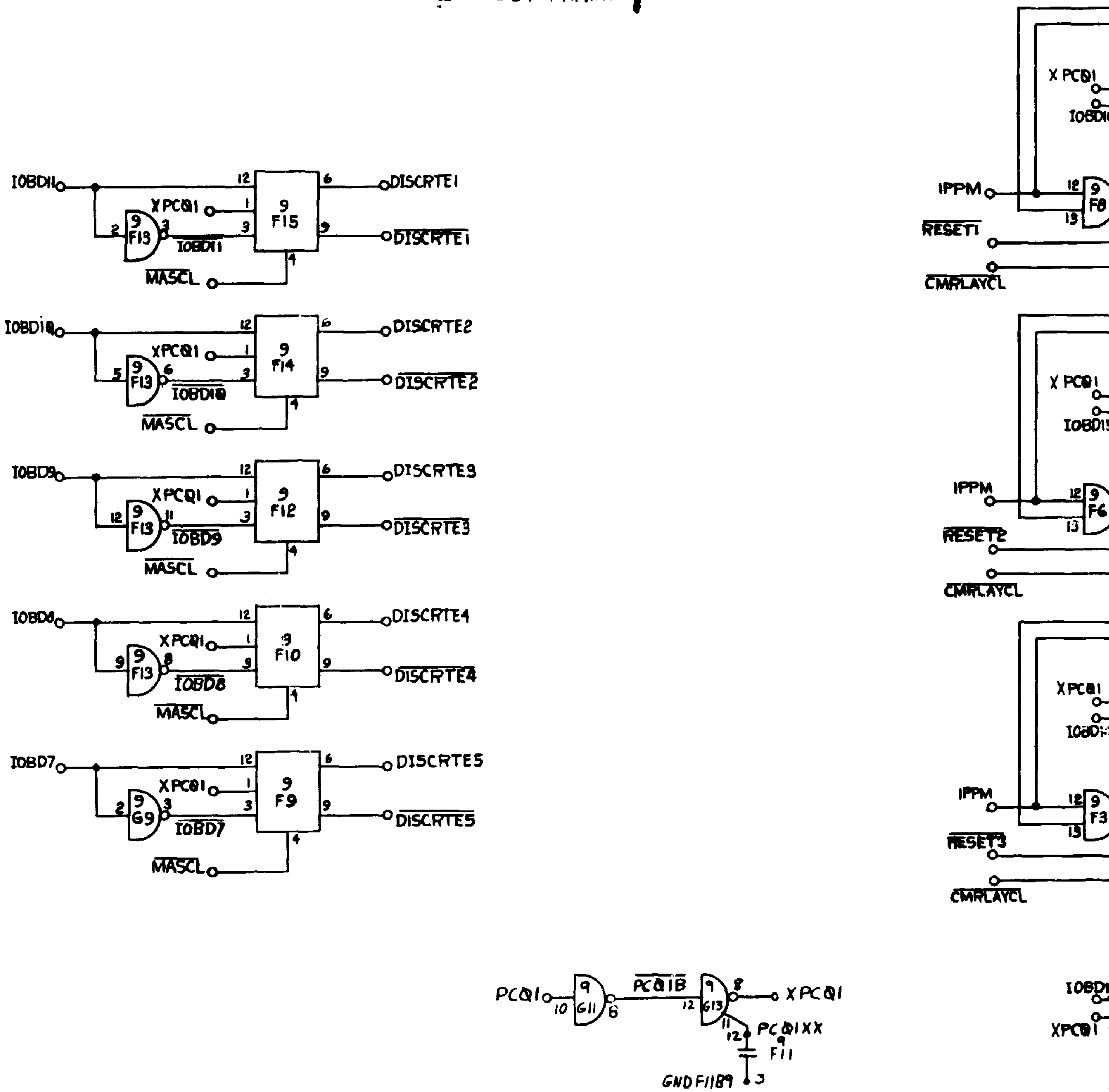


Figure A16 (GD1308599)—Logic diagram of command decoding controls.

PRECEDING PAGE BLANK NOT FILMED

FOLDOUT FRAME 1



Figu

FOLDOUT FRAME 1

FOLDOUT FRAME 2

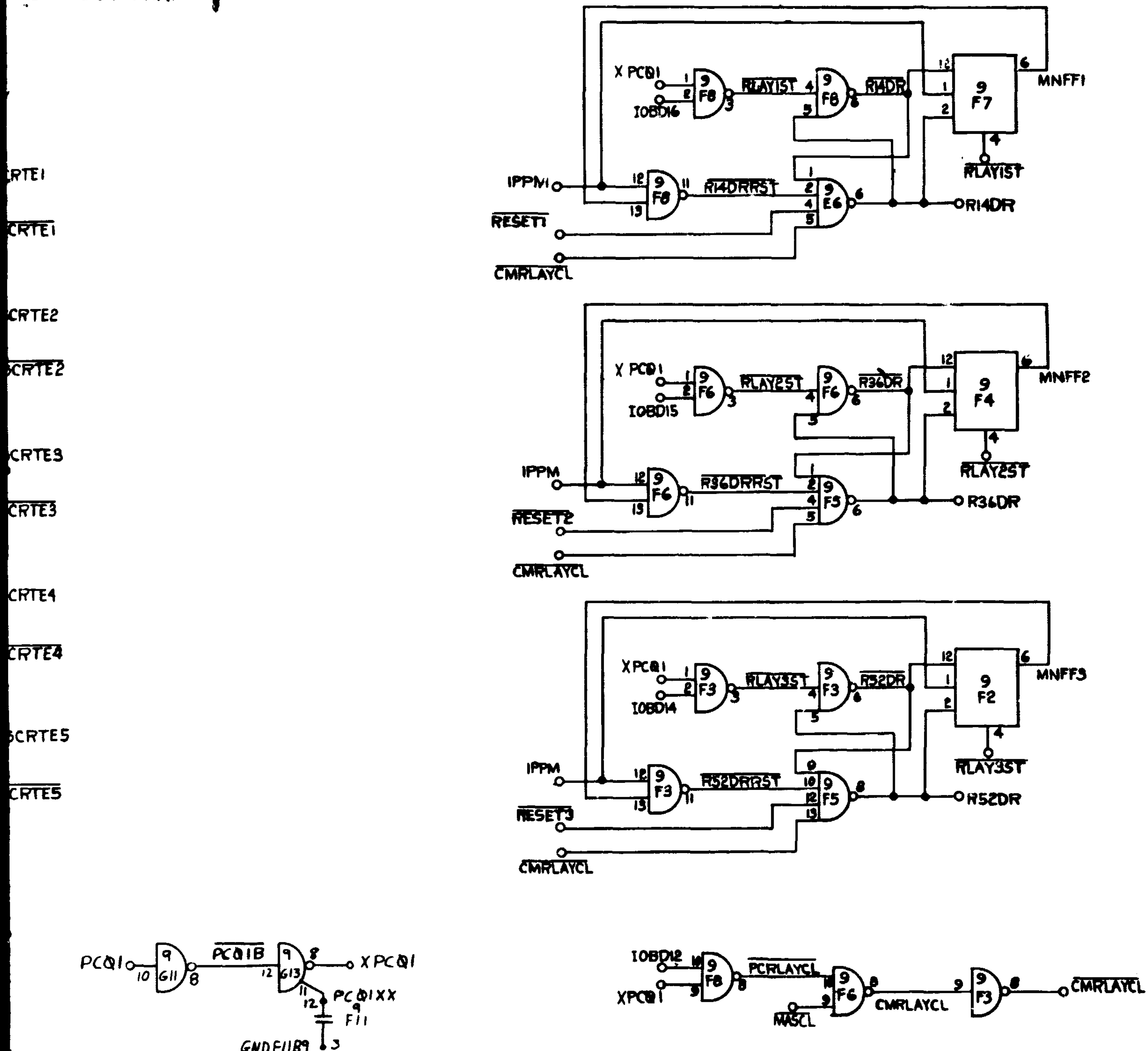


Figure A17 (GD1308602)—Discrete and relay latch logic diagram.

PRECEDING PAGE BLANK NOT FILMED

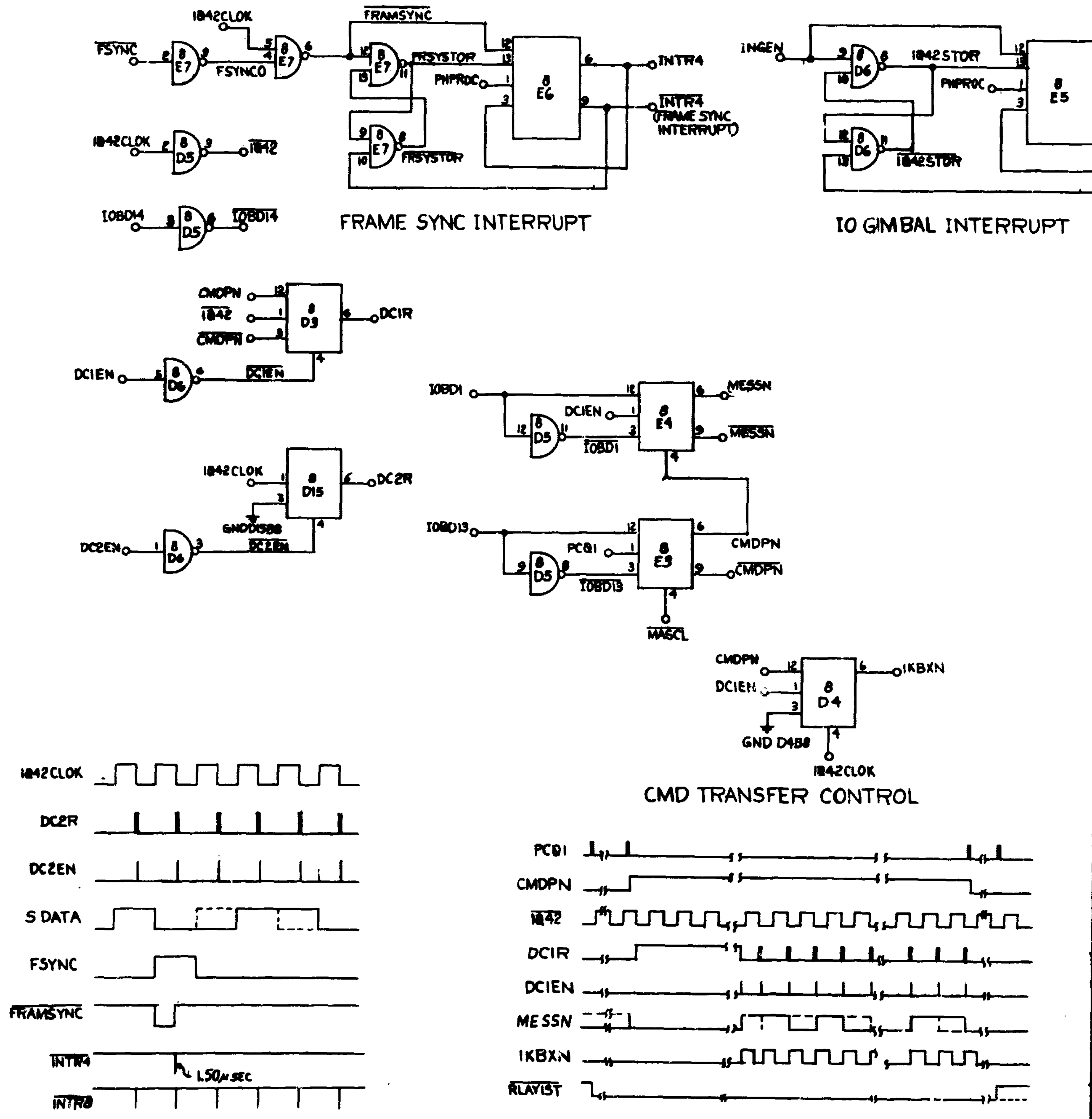


Figure A18 (GD1308689)—Logic diagram of Interrupts 4 and 8, and contro

NK NOT FILMED

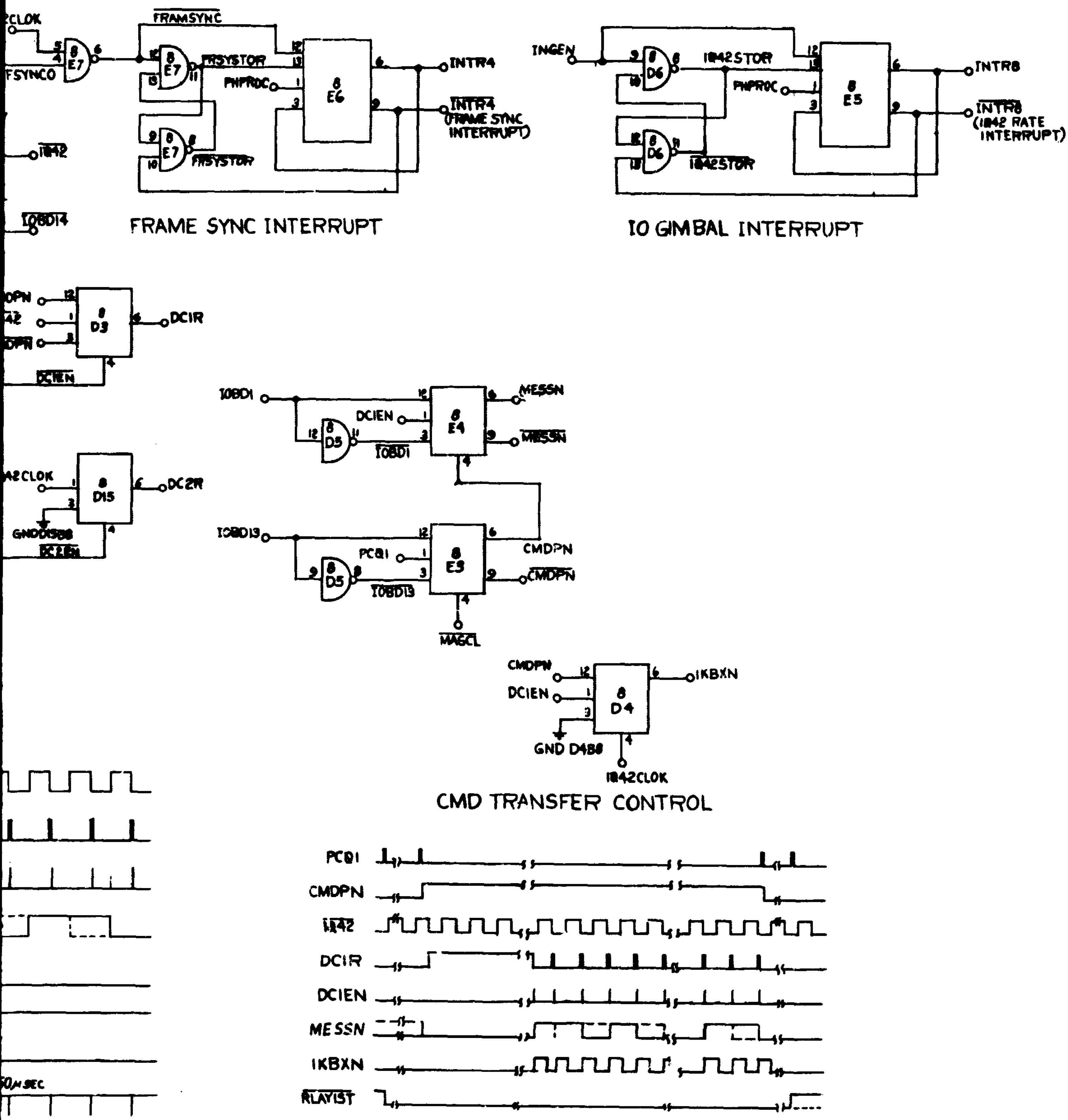
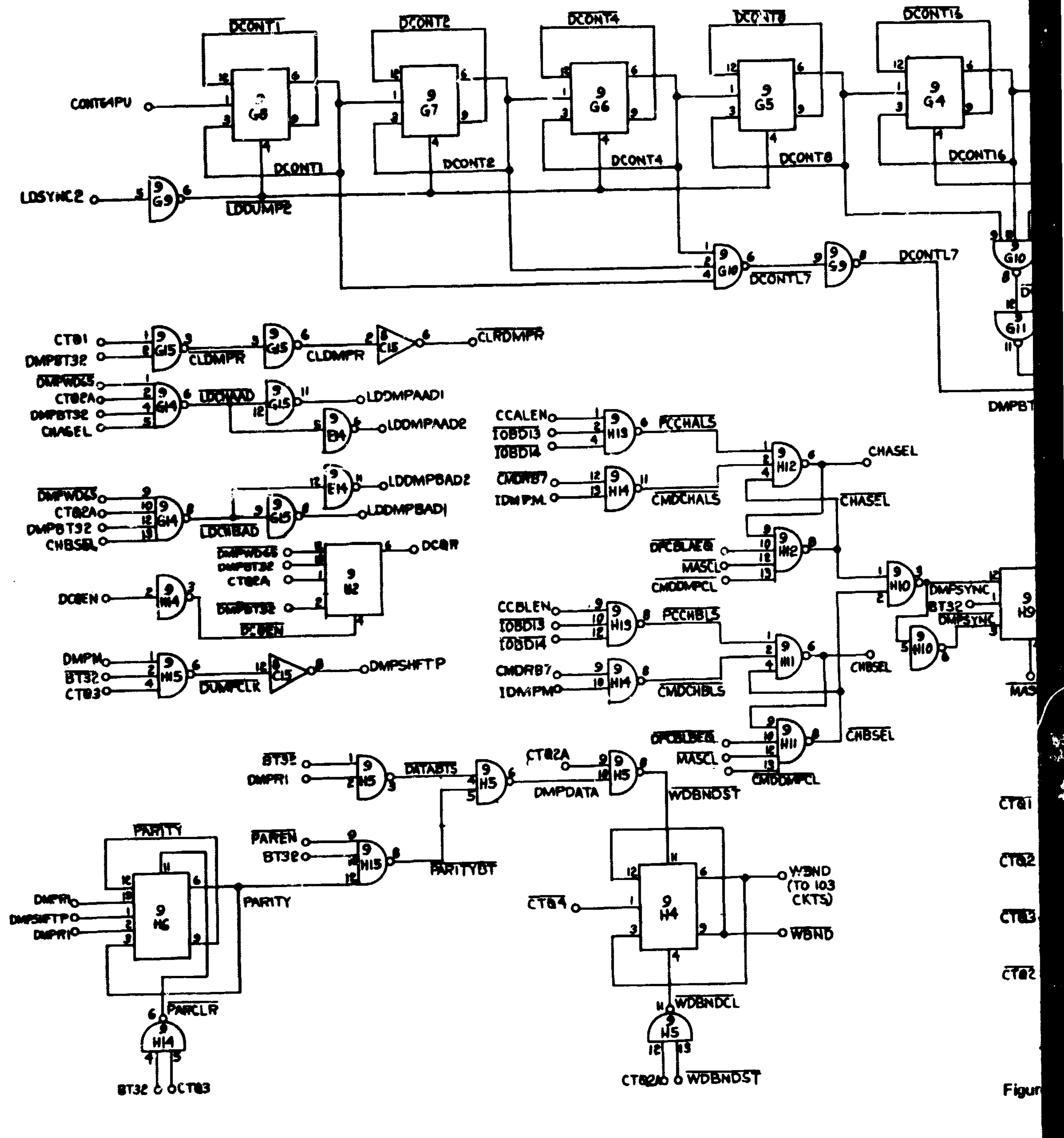


Figure A18 (GD1308689)—Logic diagram of Interrupts 4 and 8, and controls for command transfer.

PRECEDING PAGE BLANK NOT FILMED



Figur

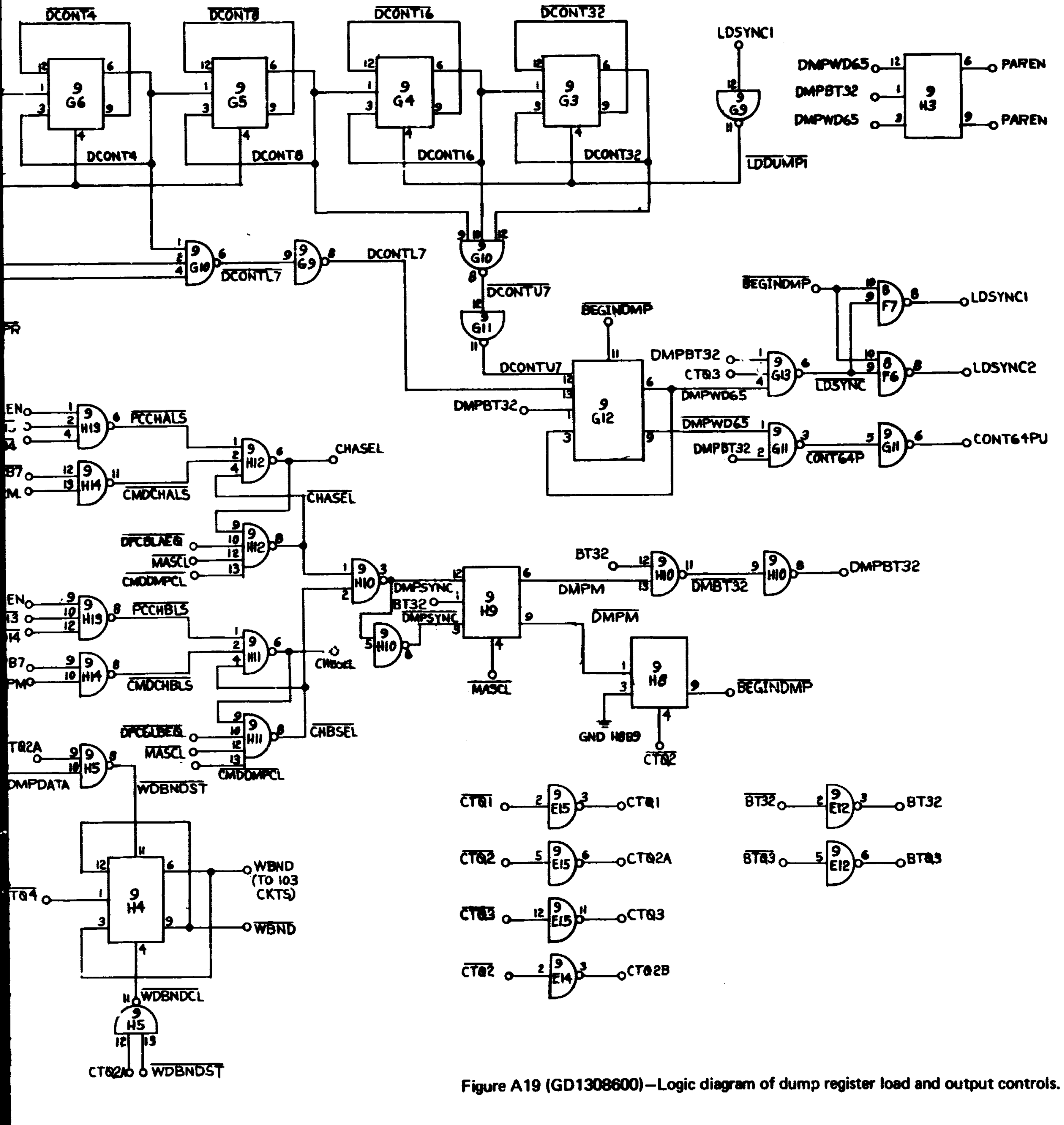
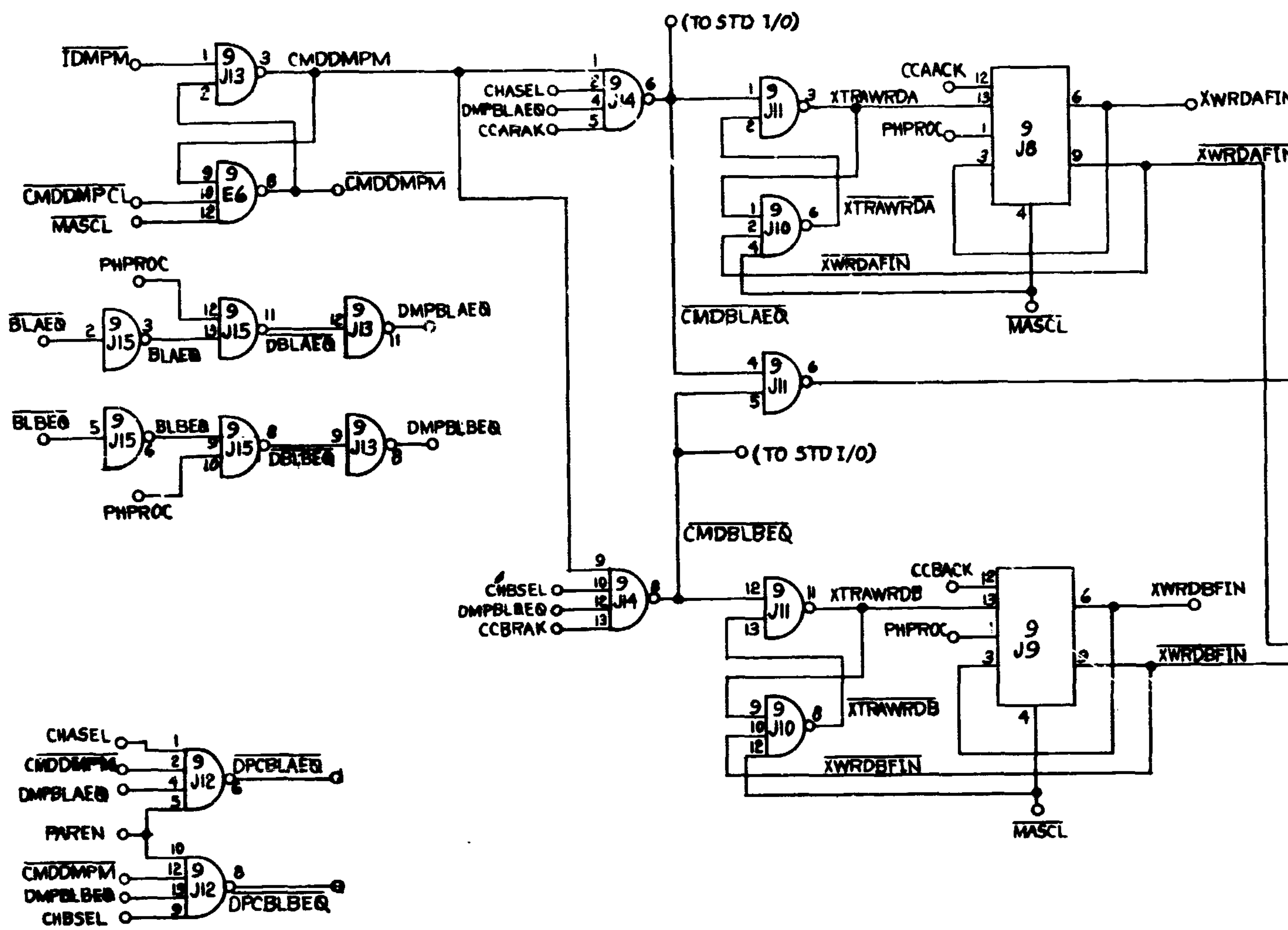


Figure A19 (GD1308600)—Logic diagram of dump register load and output controls.

PRECEDING PAGE BLANK NOT FILMED



Figu

FOLDOUT FRAME

1

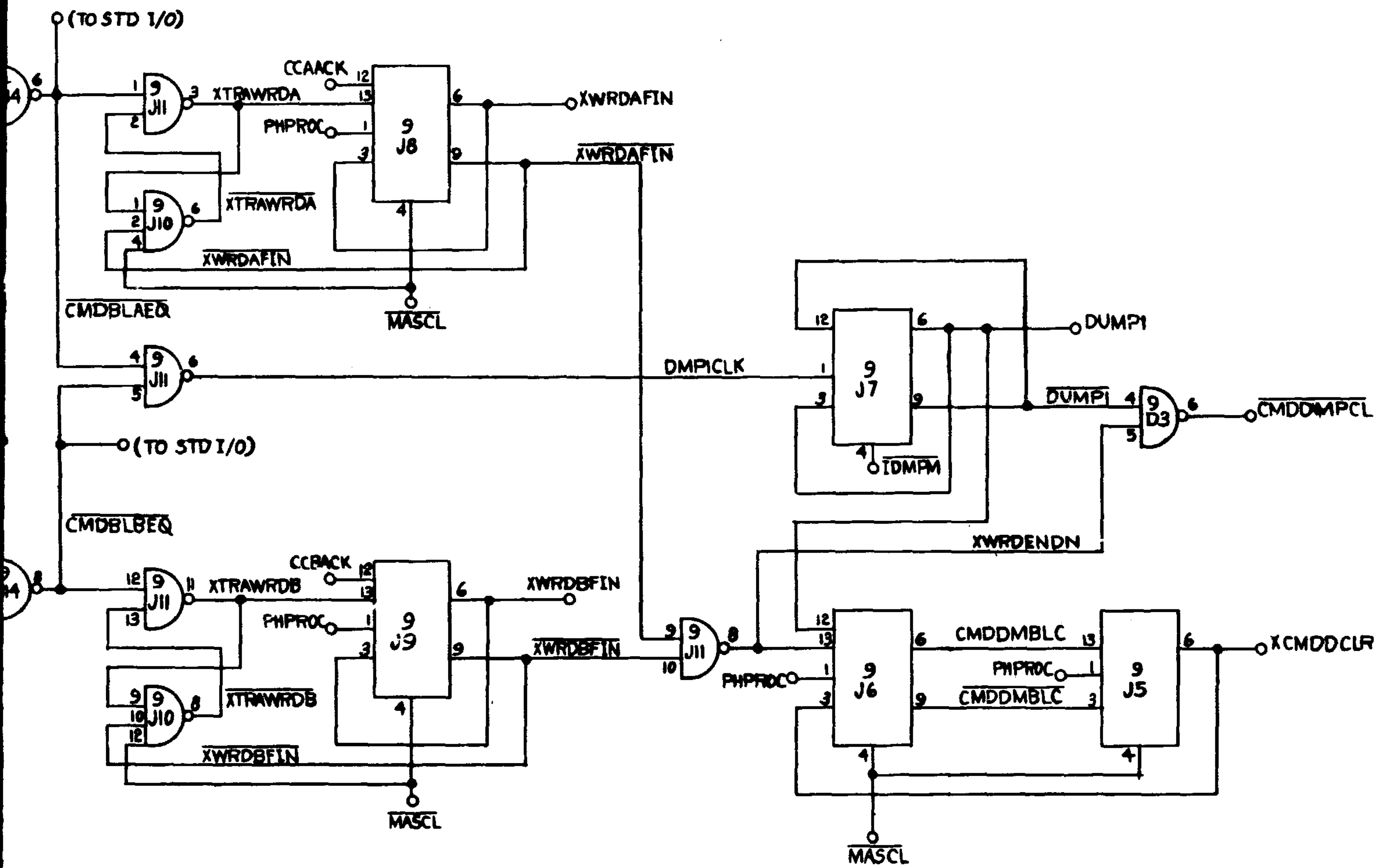
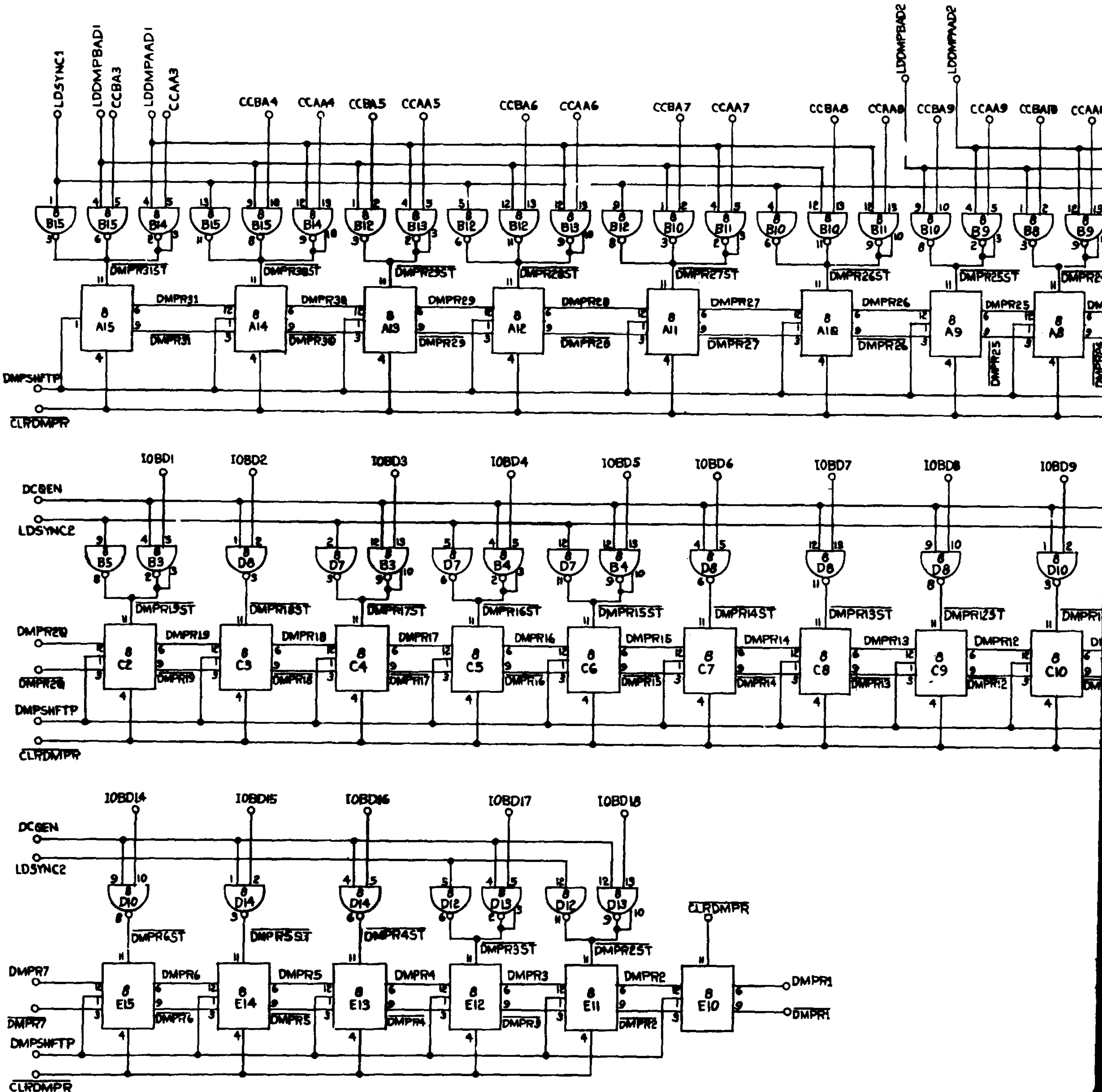


Figure A20 (GD1308598)—Logic diagram of command dump extra-word control.

FOLDOUT FRAME

2

PRECEDING PAGE BLANK NOT FILMED



FOLDOUT FRAME

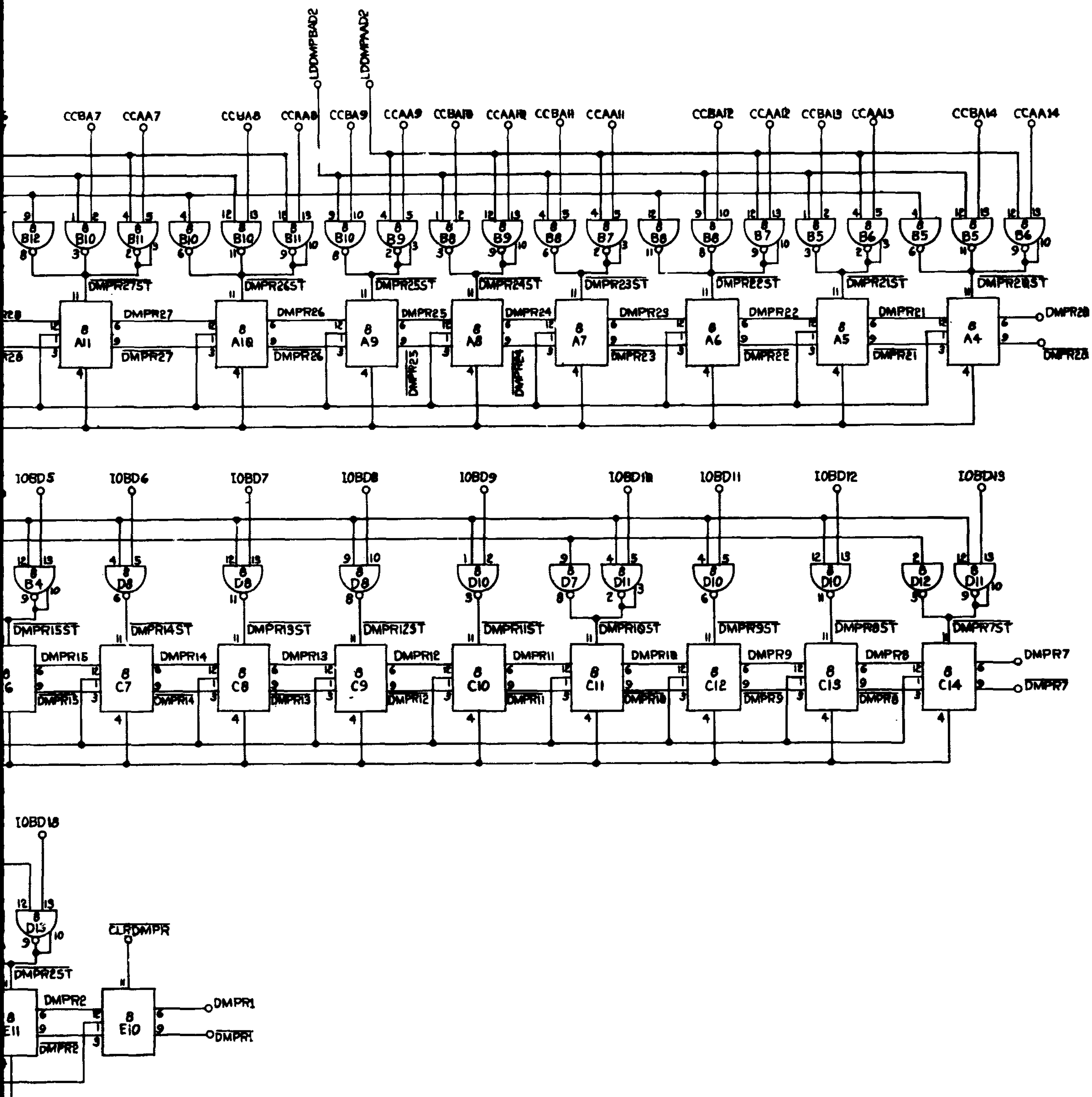
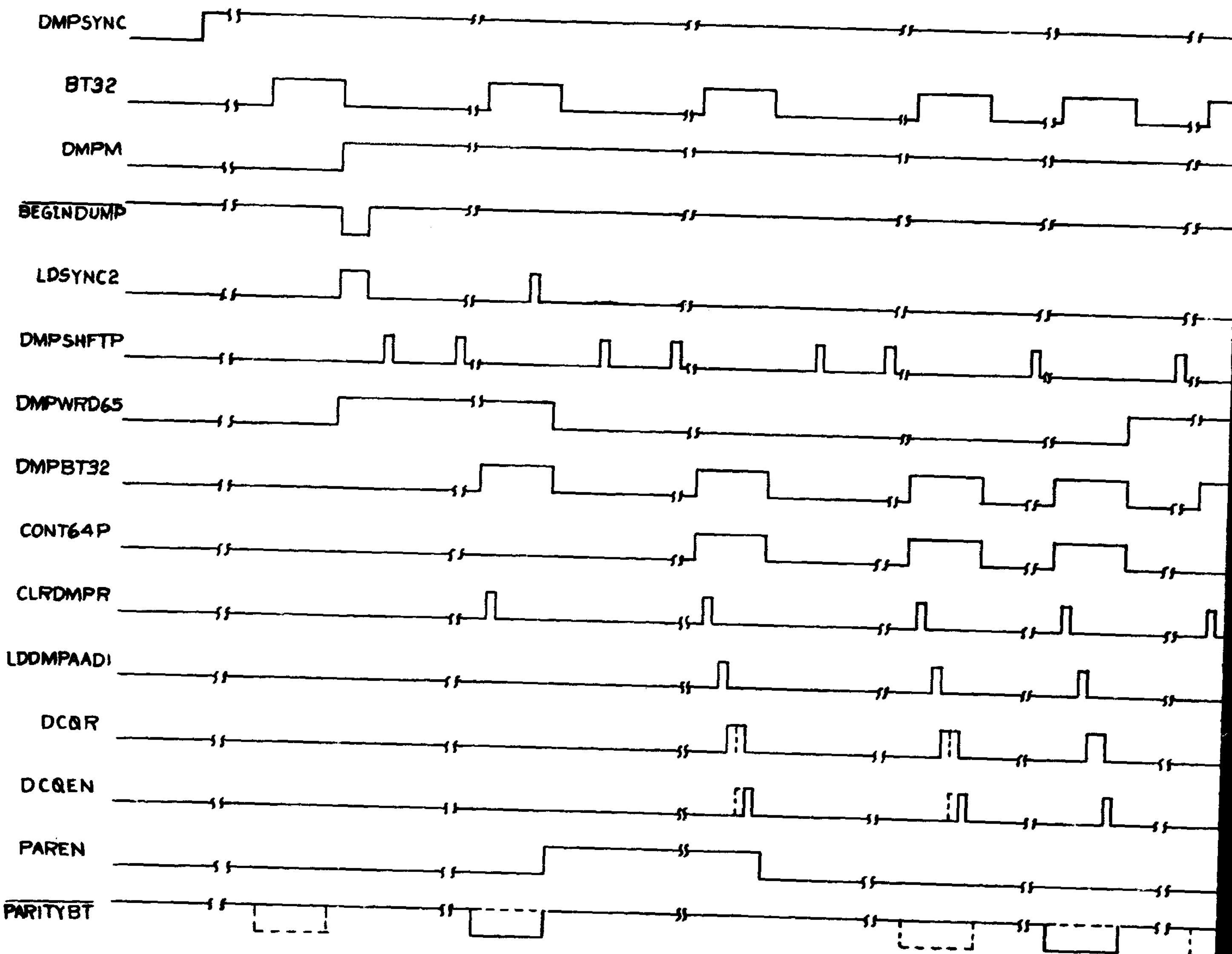


Figure A21 (GD1308595)—Dump shift register logic diagram.

FOLDOUT FRAME 2



FOLDOUT FRAME

1

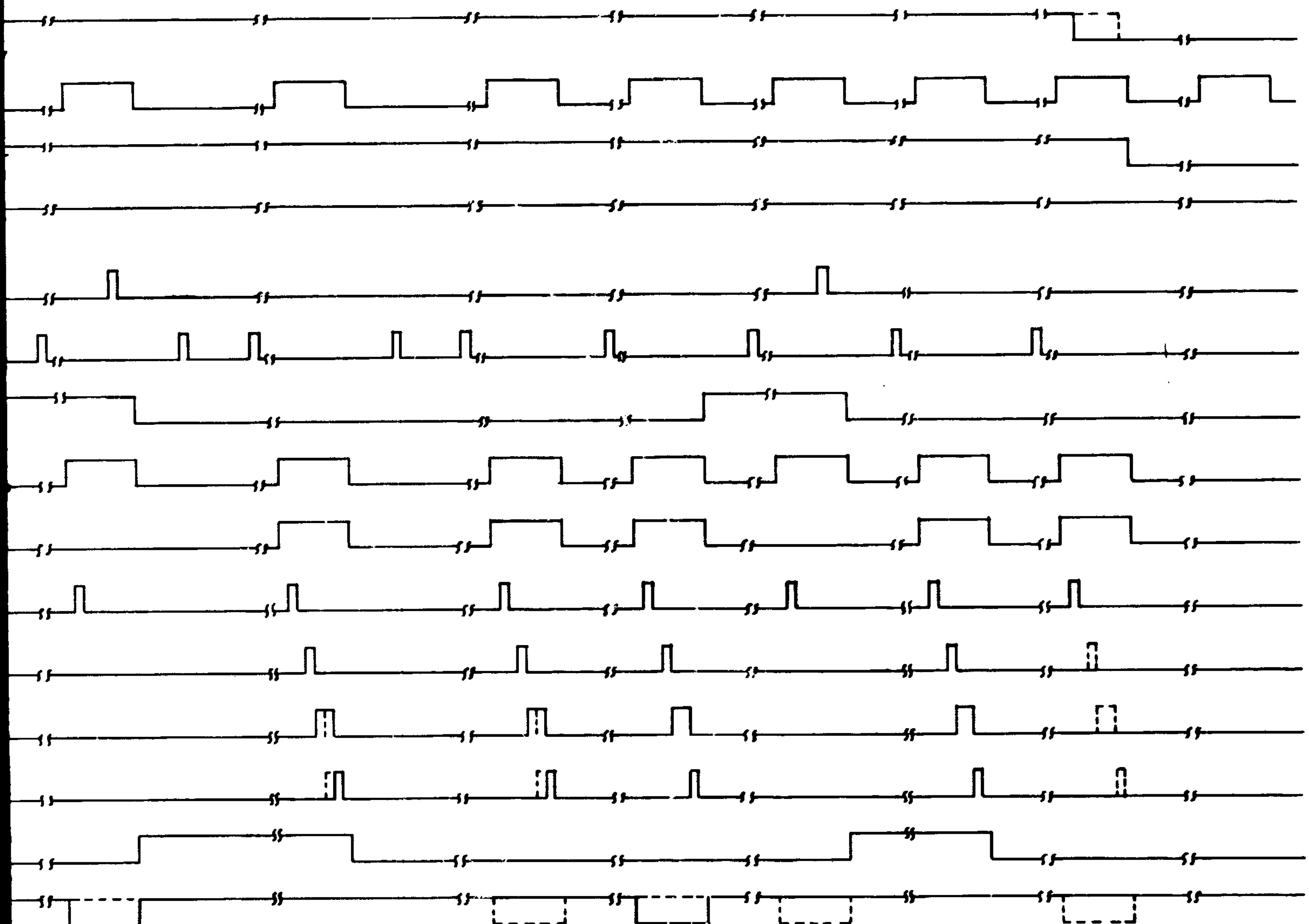


Figure A22 (GD1308660)—Dump control timing diagram.

2

PRECEDING PAGE BLANK NOT FILMED

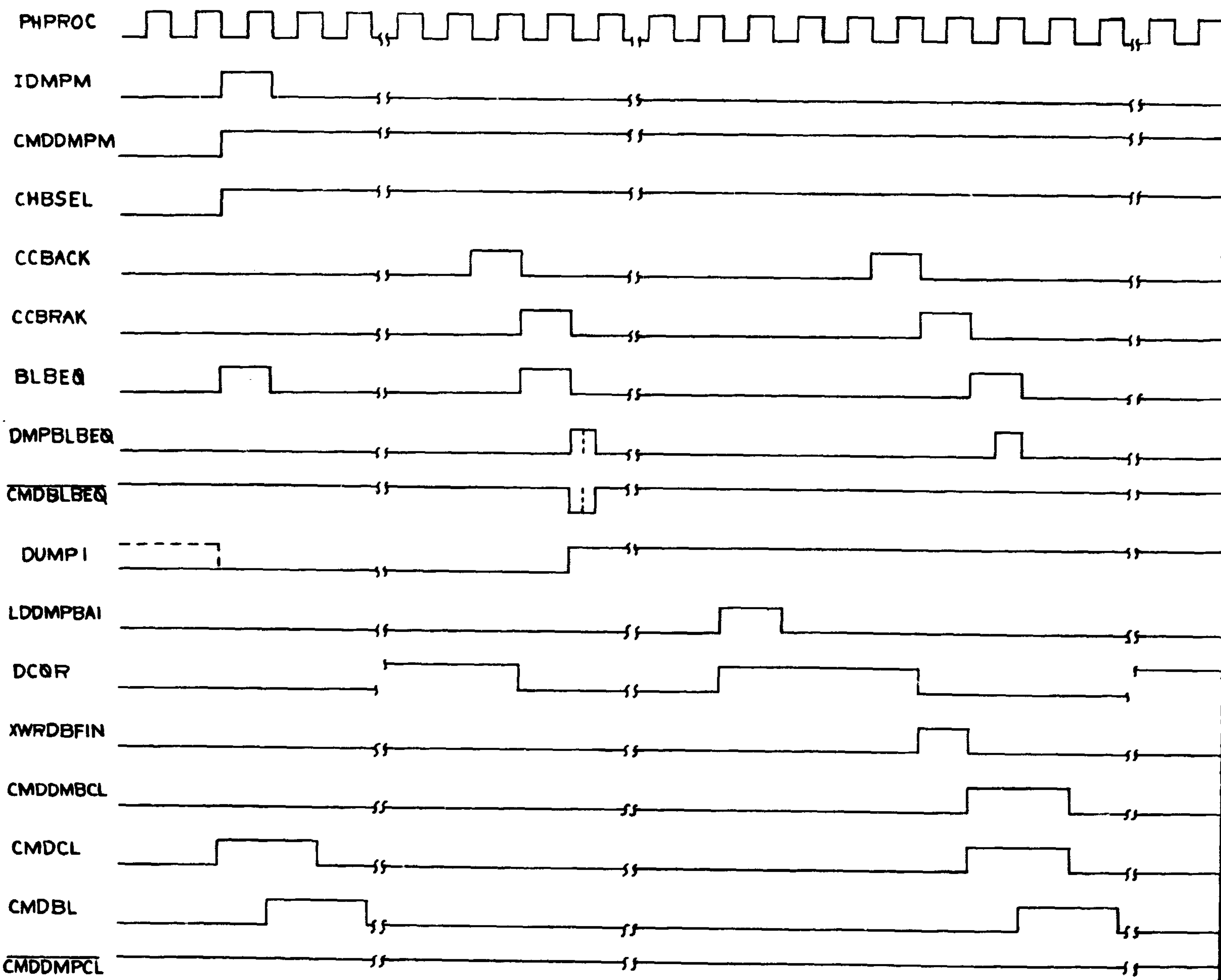


Figure A23

FOLDOUT FRAME

1

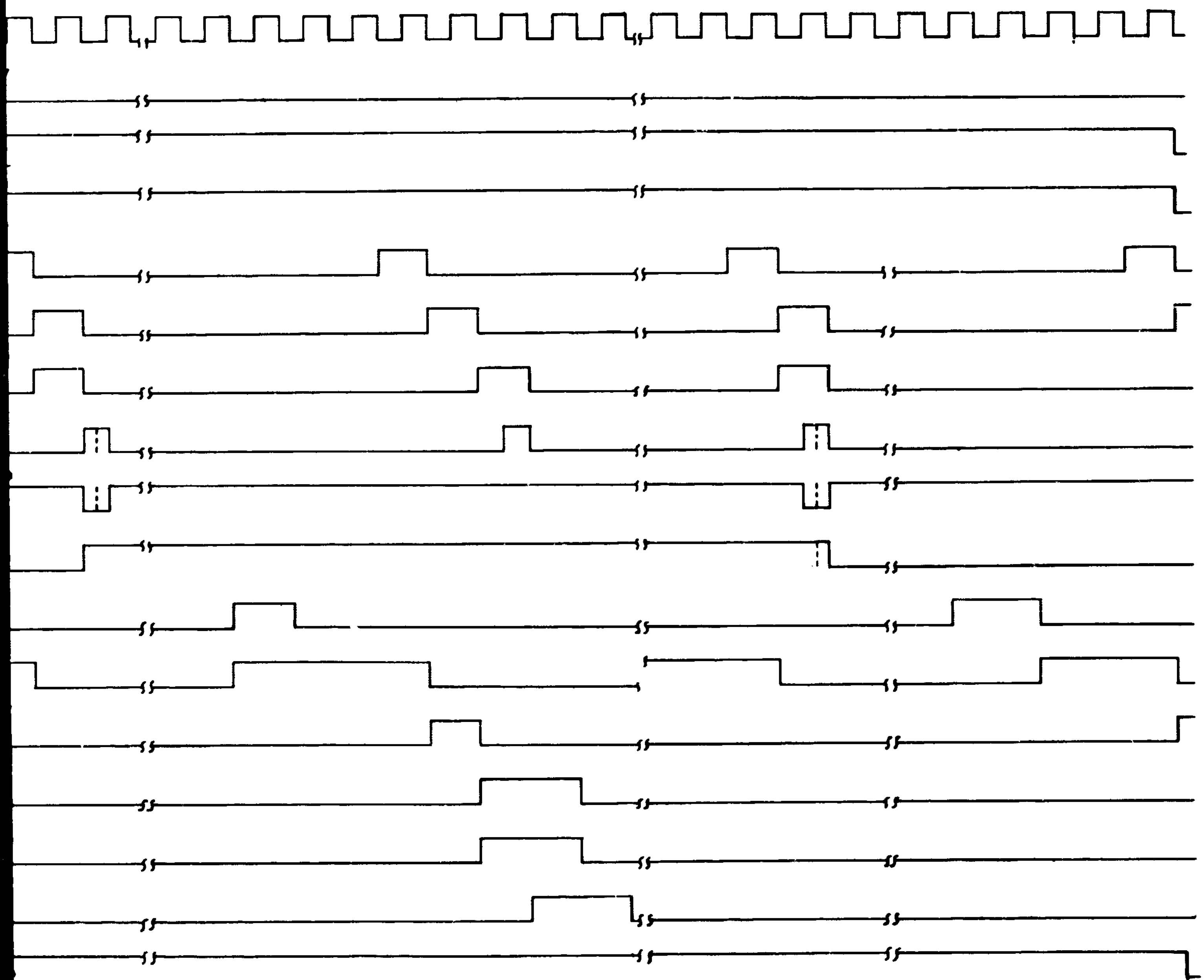


Figure A23 (GD1308658)—Command dump control timing diagram.

FOLDOUT FRAME

2

APPENDIX B

FORMATS

COMMAND FORMATS

The two experiment command channels requested by the OBP are used to load OAO experiment commands into the OBP memory, to control the operation of the computer, and to send code and data to the computer. The command formats used by the ACMU on OAO 2 are used on OAO C for auxiliary command storage. The only variation between formats for the OAO 2 ACMU load and an OAO C OBP auxiliary command load is that on OAO 2, bits 23 through 28 are "don't care" bits in the experiment internal command word for ACMU commands, and on OAO C, these bits must become octal 40.

The OBP has been assigned experiment command channels 1 and 2 for the OAO C flight. Table B1 presents all command formats, and Table B2 shows the formats of code words used by the operational program. Experiment command channel 2 is used only to hold the second half of an auxiliary command to be stored. This line cannot be used for other functions, because the second half of an auxiliary command uses all 30 bits. Bit 3 must always equal one for experiment command channel 1 because it is used for registration purposes. Bit 32 of experiment commands received on channel 1 is used to distinguish between the first half of an auxiliary command and all other OBP commands. A zero in bit 32 indicates that bits 4 through 31 on experiment command channel 1 are the 28 bits of the first half of an auxiliary command. As seen in Table B1, if bits 3 and 32 on experiment command channel 1 are both ones, bits 23 through 28 become a 6-bit function code that is decoded by the I/O unit. Depending on the function code, the 18-bit word in positions 5 through 22 is either used by the I/O unit or the CPU, or ignored.

The first two OBP commands listed on Table B1 have I/O function codes of 01 and 02. This command pair, issued consecutively in that order, results in initiation of a CSC in the I/O unit for a block data transfer with memory. The first command with an octal 01 for an I/O function code contains a device number (0 through 3), a CSC (A or B), and a block length (0 through 4096) in the 18-bit field. The next command (octal 02) contains a 14-bit starting address. The control for this block transfer is commanded from one of two redundant CSC's in the I/O unit. In the true sense, cycle-steal redundancy exists because the assignment of the CSC is independent of data source/receiver. To load new programs or data from a ground station, an 01/02 command pair must be sent on experiment command channel 1 followed by the new data to be stored. The new data are in the form of one OBP word per experiment command 1 with an octal 30 function code. Thus, to load the numbers 0, 1, 2,

and 3 into OBP memory locations 10, 11, 12, and 13 (octal) would require the following command sequence:

3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	X	X	X	1
1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	X	X	X	1
1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	X	X	X	1
1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	X	X	X	1
1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	X	X	X	1
1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	X	X	X	1

The function code 04 in bits 23 through 28 of an experiment command 1 selects the bank of memory in which "fixed locations" used for I/O and interrupts are to be stored. Since this function can be reassigned to another memory bank, it provides for continued operation even if the memory bank containing "fixed locations" were to fail. Bank 0 normally contains the "fixed location" information, but this could be transferred to another bank; of course, some program change and reload would then be required. Function code 07 is a master clear that resets both the I/O unit and the CPU. Function codes 10 through 13 cause a memory dump of banks 0 through 3 to the wideband transmitter through CSCA, and codes 14 through 17 cause a dump of banks 0 through 3 through CSCB. Function code 20 causes interrupt 1 to occur. This interrupt is used to initiate certain data locations, clear off all interrupts, and pass control to the executive program; it can be commanded after a new program is loaded and occurs automatically after recovery from under voltage on board. Function code 40 initiates an interrupt 3 request. This interrupt program executes an input instruction to bring in bits 5 through 22 of the command 1 and takes appropriate action after deciphering bits 5 through 9 of the input word. Codes that have been defined are given in Table B2. Note that bits 5 through 9 are also ACMU internal command bits on OAO 2 so that one "appropriate action" might be to respond to an ACMU command. Command type 40's can also be used to initiate block transfers under software control rather than initiating the CSC with command types 01 and 02.

Memory dumps can also be initiated under program control by using a command type 40. It is seen that other functions can be defined for type-40 commands since these commands result in an interrupt request with a 5-bit code and 13-bit program words available for use by the operational program in the OBP.

Dump Format

Upon command, the OBP dumps a selected bank of memory twice to the wideband transmitter. The serial bit rate of the dump is 50 kHz, and the data are converted to split-phase prior to transmission. A dump transmission requires approximately 5.32 s, since a bank is dumped twice and the format is 32 bits per word, 65 words per frame, 64 frames per dump. Each 32-bit word is formatted as follows:

1	2	19	20	31	32
1	18-bit OBP word	MSB	12 MSB's of memory address MSB		P

It should be noted that a full address word is 14 bits; hence, the 12 most significant address bits increment by 1 every four words. Bits 20 and 21 can be used to determine which memory bank is being dumped. Bit 19 of the dump word corresponds to bit 1 of an OBP memory word, and bit 2 of the dump word holds bit 18 of the word readout of OBP memory. This arrangement differs from the manner in which the 18 bits are formatted for an ACMU dump on OAO 2: To conserve hardware, the 18-bit word is split into a 3-bit and a 15-bit field on ACMU.

Table B1 – Command code formats.

Bits 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Note	Channel	Format
Note 1	Experiment Command Channel 1	1 | 28-bit first half of command to be stored
Note 1	Experiment Command Channel 2	30-bit second half of command to be stored
	Experiment Command Channel 1	1 | X | 18-bit word | I/O FN code |

Note	Function (FN)	03	04	use of 18-bit word	I/O FN code (octal)	29	30	31	32
Note 1	Set block, length, channel and device for block transfer	1	X	X X CH X DEVICE MSB block length LSB	01	X	X	X	1
Note 1	Set starting address for block transfer	1	X	X X X X MSB starting address LSB	02	X	X	X	1
	Set "fixed location" bank	1	X	X X X X WHICH BANK	04	X	X	X	1
	Dump memory bank 0, Ch A	1	X	Not used	10	X	X	X	1
	Dump memory bank 1, Ch A	1	X	Not used	11	X	X	X	1
	Dump memory bank 2, Ch A	1	X	Not used	12	X	X	X	1
	Dump memory bank 3, Ch A	1	X	Not used	13	X	X	X	1
	Dump memory bank 0, Ch B	1	X	Not used	14	X	X	X	1
	Dump memory bank 1, Ch B	1	X	Not used	15	X	X	X	1
	Dump memory bank 2, Ch B	1	X	Not used	16	X	X	X	1
	Dump memory bank 3, Ch B	1	X	Not used	17	X	X	X	1
	Interrupt 1 (Initiate)	1	X	Not used	20	X	X	X	1
	Data request	1	X	18-bit data word to be stored in OBP memory	30	X	X	X	1
Note 2	Interrupt 3 (code word)	1	X	5-bit code | 13-bit program word	40	X	X	X	1

Note 1: These two commands must be issued consecutively in the order shown.
Note 2: See Table A2 for code formats.

Table B2–Program code formats.

03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Experiment Command Channel 1					
Function (FN)	1 X	5-bit code	13-bit program word	I/O FN code	X X X 1
Clear Page to 0's	1 X	0 0 0 0 0	Page / Not Used	1 0 0 0 0 0	X X X 1
Set Page to 1's	1 X	0 0 0 0 1	Page / Not Used		
Initiate Page, erase	1 X	0 0 0 1 0	Page / Not Used		
Initiate Page, no erase	1 X	0 0 1 0 0	Page / Not Used		
Initiate Page, set RT mode	1 X	1 1 0 1 0	Page / Not Used		
Transfer Page to PPDS	1 X	1 0 0 0 0	Page / Not Used		
Set relay latch 1	1 X	1 0 0 0 0	Not Used		
Set relay latch 2	1 X	1 0 0 1 0	Not Used		
Set relay latch 3	1 X	1 0 1 0 0	Not Used	1 0 0 0 0 0	X X X 1

Note: Above functions are related to auxiliary command memory.

Dump bank to wideband	1 X	1 1 0 0 0	Bank / Not used	1 0 0 0 0 0	X X X 1
Initiate CSC and set up 1-word buffer for SA	1 X	0 1 0 0 0	X / Block length LSB	1 0 0 0 0 0	X X X 1
FN code to programs	1 X	0 1 0 1 1	13-bit code to executive	1 0 0 0 0 0	X X X 1

APPENDIX C

DETAILED DESCRIPTION OF CENTRAL PROCESSOR UNIT

The CPU block diagram shown in Figure C1 illustrates the information flow through the various registers and logic units.

The abbreviations used in the logic flow charts and the number of bits contained in each register are noted within each box, the number of bits appearing in parentheses. Blocks containing no abbreviation or bit count are considered logic units. Lines with double arrows indicate that information can flow in either direction between two registers. Heavy lines indicate a bus.

In general, data transfer is parallel. The major exception to this rule is the Accumulator (A) and the Extended Accumulator (EA), where several complex two-way serial data paths exist as required by the instruction set.

DESCRIPTION OF REGISTERS

Accumulator

The Accumulator, an 18-bit register, contains one operand for all arithmetic, shifting, input/output, and logic functions and is used to retain the results of all arithmetic, shifting, and logic functions

Address Register

The Address Register (ADR) is an 18-bit register used to hold the memory address information for all CPU memory cycles. Although only 14 bits are required for addressing the OBP-2 16K memory, the register is full size in order to allow basic system expansion to 64K or more and also because it is used to hold the 18-bit divisor during the divide instruction.

Extended Accumulator

The EA is an 18-bit register that provides the required storage in the multiply and divide operations. The remainder is stored in the EA after a divide. After a multiply, the low half of the product is stored in the EA. In addition to these functions, the Accumulator with the EA can be normalized and shifted both cyclically and algebraically by means of the appropriate instructions.

Subscript Register

The 18-bit Subscript Register (SS) can be loaded, stored, added to, incremented, and tested by the SS instruction. It is used by subscripted (index) instructions to modify the operand address by adding in the value contained in this register during the formation of an effective operand address (see "Instruction Execution"). Its ready accessibility makes it available for general use.

Scale Register

The 6-bit Scale Register (SCALE) can be loaded and stored by program control. Its function is to denote the position of the binary point associated with a data word. This register makes normalized fractional arithmetic possible by using a setting of zero, and integer arithmetic possible by using a setting of 17. Other settings between −32 and +31 allow a wide, dynamic range for binary point placement.

Storage Limit

The 18-bit Storage Limit Register (SLR) is divided into two fields of nine bits each. Its purpose is to permit storage in memory only within the range specified for the program. This register can be modified only from the interrupt cells located at the beginning of memory.

Page Register

The 4-bit Page Register (PAGE) is appended to the address field of all instructions and determines in which bank of 4096 words the address field is located. If indexing is used, it can modify the effective address to the point of changing the addressed bank.

Instruction Register

The 11-bit Instruction Register (IRS) is used to store the operation code and the subscript bit of the instruction. The output of this register is decoded to determine the instruction type and remains unchanged throughout the execution of that instruction.

Carry Register

The Carry Register (C) is a 1-bit register that is changed by operations such as addition, subtraction, and negation. The setting condition is a carry out of the 17th bit of the two's complement adder. If the setting condition is not met, a zero is placed in the Carry Register.

Overflow Register

The 1-bit Overflow Register (OV) is set by arithmetic and shifting instructions if overflow has occurred. (Overflow is defined as the loss of a significant bit out of the 17th bit of the Accumulator.) This register is reset only by testing or by the instruction to reset it.

Instruction Counter

The 16-bit Instruction Counter (IC) contains the memory address of the instruction under execution and performs the updating of this address during all nonbranching instructions. The IC is loaded with the 16 low-order bits from storage upon execution of a branch instruction or an interrupt.

Memory Operand Register

The 18-bit Memory Operand Register (MOR) is the memory data bus interface register for the processor. It provides all data to and receives all data from the memory. In addition to these functions, the MOR holds one operand for the processor in all arithmetic and logic operations.

Or/And Register

The 1-bit Or/And Register (OA) is cleared or set by the *OR* and *AND* instructions, respectively, and is used to determine the setting of the Decision Register (D) when one or more subsequent test conditions are specified. If the OA is a one, an *AND* condition is specified; therefore, both "prior test true" (D = 1) and "present test true" are required to set D. If the OA is a zero, an *OR* is specified; therefore, "prior test true" or "present test true" will set D. At the conclusion of the conditional transfer, or upon the execution of an *OR* instruction, the OA is reset to zero. The execution of an *AND* instruction will set the OA to one.

Decision Register

The 1-bit Decision Register (D) is set by the test instructions if the proper conditions as mentioned above are met. If D is a one, the conditional transfer instruction will perform the branch, and D will be set to zero. A "reset D instruction" is also included.

Operation Counter

The 6-bit Operation Counter is required for the multiply, divide, and shift type of instructions. The counter may count either up or down and contains a value equal either to the shift length for a shift instruction or to the Scale Register for the multiply or divide instruction.

INSTRUCTION EXECUTION

Instructions are stored along with data in the memory of the OBP. The Instruction Counter contains the address of the instruction to be executed. The instruction is fetched from storage into the Instruction Register. The five high-order bits (bits 14 to 18) are used to determine the major operation code. A major *OP CODE* of zero is a nonmemory access instruction, and the operation to be performed is determined by the minor *OP CODE*

located in bits 1 to 5, as illustrated in Figure C2. While performing the instruction decode, the Instruction Counter is increased by one in order to point to the next instruction to be executed.

All other major *OP CODE*'s result in a memory access because they require that an operand be fetched. The 16-bit memory address is formed as shown in Figure C3. To establish an effective address, the content of the Page Register is appended to bits 1 through 12 of the instruction.

If subscripting is indicated (i.e., if bit 13 of the instruction is a one), the content of the Subscript Register is added to form the effective address. Once the effective address is computed, the operation indicated by the content of bits 14 to 18 is performed. Depending on the operation, the Instruction Counter is either increased by one or replaced by the content of the memory word pointed to by the effective address.

Upon completing the execution of one instruction, the processor automatically proceeds to the next. A description of all basic machine operations is presented in the next section.

MACHINE VERBS

The following list of machine operations presents a simple description of the execution for each instruction. The English name for the operation is a so-called verb to reflect the English-like language originally planned to be used to program the OBP; however, in view of the fact that these are really machine operations, a corresponding set of three-letter mnemonics has also been assigned. The memory access operations are denoted by *noun* following the operation name. This indicates that the programmer must specify the name of an operand for this verb. This name can be any word or group of words except a legitimate OBP-1 verb, e.g., *OP CODE.*

The instruction format and octal representation are shown in the box to the right of the programming language specification. The bit pattern of instruction words is shown in Figure C2. The second instruction format, if given, is the octal representation that is produced when the configuration (*verb*) *SUBSCRIPTED* (*noun*) is used in the programming language. The use of subscripting requires 4.00 μs more than the indicated execution time of the instruction.

Load and Store Instructions

LET noun, IF noun	2 0	n o u n
LET SUBSCRIPTED noun, IF SUBSCRIPTED noun	2 1	n o u n

The content of storage at the effective address is placed in the Accumulator.

Registers altered: Accumulator

Timing: 10.0 μs, 14.0 μs if subscripted

LET LOCATION OF noun	4 0 | n o u n
LET LOCATION OF SUBSCRIPTED noun	4 1 | n o u n

The effective address is placed in the Accumulator.

Registers altered: Accumulator

Timing: 8.0 μs; 12.0 μs if subscripted

YIELD noun, REPLACE noun	6 0 | n o u n
YIELD SUBSCRIPTED noun, REPLACE SUBSCRIPTED noun	6 1 | n o u n

The content of the Accumulator is stored at the effective address unless that address is protected by the Storage Limit Registers. If storage is protected, no write into memory occurs.

Registers altered: none

Timing: 12.0 μs, 16.0 μs if subscripted

SET EXTENSION WITH noun	5 2 | n o u n
SET EXTENSION WITH SUBSCRIPTED noun	5 3 | n o u n

The content of storage at the effective address is placed in the Extended Accumulator.

Registers altered: Extended Accumulator

Timing: 10.0 μs, 14.0 μs if subscripted

SAVE EXTENSION IN noun	1 0 | n o u n
SAVE EXTENSION IN SUBSCRIPTED noun	1 1 | n o u n

The content of the Extended Accumulator is stored at the effective address unless that address is protected by the Storage Limit Registers. If storage is protected, no write into memory occurs.

Registers altered: none

Timing: 12.0 μs, 16.0 μs if subscripted

Arithmetic Instructions

PLUS noun	0 4 | n o u n
PLUS SUBSCRIPTED noun	0 5 | n o u n

The content of storage at the effective address is added to the content of the Accumulator, and the sum is retained in the Accumulator. If a carry occurs at the input of the 18th stage of the two's complement adder, the Carry Register is set to one. Otherwise, the Carry Register is reset to zero. Overflow can occur when two numbers of the same sign are added. Overflow causes the 18th bit of the sum to remain in the sign position and the Overflow Register to be set to one.

Registers altered: Accumulator
Carry Register
Overflow Register (conditionally)

Timing: 12.0 μs, 14.0 μs if subscripted

MINUS noun	2 4 | n o u n
MINUS SUBSCRIPTED noun	2 5 | n o u n

The content of storage at the effective address is subtracted from the content of the Accumulator, and the difference is retained in the Accumulator. Subtraction is performed by using the one's complement of the content of storage and adding it to the Accumulator with a carry forced into the low-order stage of the adder. If a carry occurs at the input of the 18th stage of the two's complement adder, the Carry Register is set to one. Otherwise, the Carry Register is reset to zero. Overflow can occur when two numbers of different sign are subtracted. Overflow causes the 18th bit of the difference to remain in the sign position and the Overflow Register to be set to one.

Registers altered: Accumulator
Carry Register
Overflow Register (conditionally)

Timing: 10.0 μs, 14.0 μs if subscripted

TIMES noun	4 4 | n o u n
TIMES SUBSCRIPTED noun	4 5 | n o u n

The content of storage at the effective address is multiplied by the content of the Accumulator. The high-order 17 bits and sign of the product are retained in the Accumulator. The low-order 17 bits and sign of the product are retained in the Extended Accumulator. The double-length product is automatically scaled by arithmetically shifting the Accumulator and the 17 bits of the product in the Extended Accumulator the number of bit positions indicated by the content of the Scale Register. The sign bit of the Extended Accumulator is not shifted. If the content of the Scale Register is negative, the shift is right and the content of the sign fills positions vacated on the left so that no overflow is possible. If the content of the Scale Register is positive, the shift is to the left, with zeros filling positions vacated on the right. The Overflow Register is set to one if the sign bit of the Accumulator is changed during the shift.

Registers altered: Accumulator
Extended Accumulator
Overflow Register (conditionally)

Timing: If the content of the Scale Register is less than 17, 37.5 μs + (number of ones in the multiplier + |scale|) X 1.5 μs; if the content of the Scale Register is equal to or greater than 17, 37.5 μs + (number of ones in the multiplier + scale − 15) X 1.5 μs. Additional 3.0 μs if subscripted.

OVER noun, DIVIDED BY noun | 6 4 | n o u n |
OVER SUBSCRIPTED noun, DIVIDED BY SUBSCRIPTED noun | 6 5 | n o u n |

The contents of the Accumulator and Extended Accumulator are automatically scaled by shifting them the number of bit positions indicated by the content of the Scale Register. The sign of the Extended Accumulator is ignored and not shifted. If the content of the Scale Register is negative, the shift is to the left, with zeros filling positions vacated on the right, and if the sign bit of the Accumulator is changed during the shift, the Overflow Register is set to one. If the content of the Scale Register is positive, the shift is to the right, and the content of the sign fills positions vacated on the left so that overflow is impossible. The scaled Accumulator and Extended Accumulator form the dividend that is divided by the content of storage at the effective address. The Overflow Register is set if the content of the Accumulator is greater than or equal to the content of storage. The signed quotient is retained in the Accumulator, and the signed remainder is retained in the Extended Accumulator. The remainder has the same sign as the dividend and has a magnitude less than the divisor.

Registers altered: Accumulator
Extended Accumulator
Overflow Register (conditionally)

Timing: If the content of the Scale Register is less than 17, 105.0 μs + $|\text{scale}| \times 1.5$ μs; if the content of the Scale Register is equal to or greater than 17, 105.0 μs + (scale − 15) × 1.5 μs; in either case, add 9.0 μs if dividend is less than zero. Additional 3.0 μs if subscripted.

PLUS CARRY | 0 0 | 0 0 | 0 6 |

The content of the Carry Register is added to the content of the Accumulator, and the sum is retained in the Accumulator. If a carry occurs at the input of the 18th bit of the two's complement adder, the Carry Register is set to one. Otherwise, the Carry Register is reset to zero. Overflow can occur and will cause the 18th bit of the sum to remain in the sign position and the Overflow Register to be set to one.

Registers altered: Accumulator
Carry Register
Overflow Register (conditionally)

Timing: 6.0 μs

NEGATED | 0 0 | 0 0 | 0 4 |

The content of the Accumulator is replaced by its two's complement. Negating all zeros yields a result of zero and sets the Carry Register to one. Negating the number that has zeros in all bit positions except the sign yields the same number as a result and sets both the Carry Register and the Overflow Register to one. Other than these two special cases, the Carry Register is reset to zero.

Registers altered: Accumulator
Carry Register
Overflow Register (conditionally)

Timing: 6.0 μs

Logic Instructions

ANDED WITH noun
ANDED WITH SUBSCRIPTED noun

3 0	n o u n
3 1	n o u n

The content of storage at the effective address is anded with the content of the Accumulator. The result is retained in the Accumulator. The 18 bits of the result are computed independently, with a one occurring in a bit position of the result only if the Accumulator and storage both contained a one in that bit position.

Registers altered: Accumulator

Timing: 7.5 μs, 10.5 μs if subscripted

OR-ED WITH noun
OR-ED WITH SUBSCRIPTED noun

5 0	n o u n
5 1	n o u n

The content of storage at the effective address is or-gated with the content of the Accumulator. The result is retained in the Accumulator. The 18 bits of the result are computed independently, with a one occurring in a bit position of the result if either the Accumulator or storage contained a one in that bit position.

Registers altered: Accumulator

Timing: 7.5 μs, 10.5 μs if subscripted

EOR-ED WITH noun (EXCLUSIVELY OR-ED WITH noun)
EOR-ED WITH SUBSCRIPTED noun (EXCLUSIVELY OR ED WITH SUBSCRIPTED noun)

7 0	n o u n
7 1	n o u n

The content of storage at the effective address is exclusively or-gated with the content of the Accumulator. The result is retained in the Accumulator. The 18 bits of the result are computed independently, with a one occurring in a bit position of the result if either the Accumulator or storage, but not both, contained a one in that bit position.

Registers altered: Accumulator

Timing: 7.5 μs, 10.5 μs if subscripted

COMPLEMENTED

0 0	0 0	1 0

The content of the Accumulator is complemented, and the result is retained in the Accumulator. The 18 bits of the result are computed independently, with a one occurring in a bit position of the result only if the Accumulator contained a zero in that position.

Registers altered: Accumulator

Timing: 6.0 μs

Bit Manipulation Instructions

SHIFTED BY noun	1 4	n o u n
SHIFTED BY SUBSCRIPTED noun	1 5	n o u n

The 6 low-order bits of the content of storage at the effective address are used as a two's complement shift count. If the count is negative, the Accumulator is shifted to the right the number of positions specified by the count, with the content of the Accumulator sign replacing vacated positions on the left. If the count is positive, the Accumulator is shifted to the left the number of positions specified by the count, with zeros filling vacated positions on the right. The Overflow Register is set to one if the sign bit of the Accumulator is changed during the shift.

Registers altered: Accumulator
Overflow Register (conditionally)

Timing: 7.5 μs + 1.5 μs per position shifted; if subscripted, 10.5 μs + 1.5 μs per position shifted

DOUBLE SHIFTED BY noun	3 6	n o u n
DOUBLE SHIFTED BY SUBSCRIPTED noun	3 7	n o u n

The 6 low-order bits of the content of storage at the effective address are used as a two's complement shift count. The Accumulator and the Extended Accumulator are shifted together. The Extended Accumulator is shifted to the right of the Accumulator, and its sign bit is not shifted. If the count is negative, the accumulators are shifted to the right the number of positions specified by the count, with the content of the Accumulator sign replacing vacated positions on the left. If the count is positive, the accumulators are shifted to the left the number of positions specified by the count, with zeros filling vacated positions on the right. The Overflow Register is set to one if the sign bit of the Accumulator is changed during the shift.

Registers altered: Accumulator
Overflow Register (conditionally)

Timing: 7.5 μs + 1.5 μs per position shifted; if subscripted, 10.5 μs + 1.5 μs per position shifted

CYCLED BY noun	3 4	n o u n
CYCLED BY SUBSCRIPTED noun	3 5	n o u n

The 6 low-order bits of the content of storage at the effective address are used as a two's complement shift count. If the count is negative, the content of the Accumulator is shifted cyclically to the right the number of positions specified by the count,

with bits leaving the low-order position entering the sign position. If the count is positive, the content of the Accumulator is shifted to the left the number of positions specified by the count, with bits leaving the sign position entering the low-order position.

Timing: 7.5 μs + 1.5 μs per position shifted; if subscripted, 10.5 μs + 1.5 μs per position shifted

DOUBLE CYCLED BY noun | 5 6 | n o u n |

DOUBLE CYCLED BY SUBSCRIPTED noun | 5 7 | n o u n |

The 6 low-order bits of the content of storage at the effective address are used as a two's complement shift count. If the count is negative, the content of the Accumulator and Extended Accumulator is shifted cyclically to the right the number of positions specified by the count, with bits leaving the low-order position of the Extended Accumulator entering the sign of the Accumulator and bits leaving the low-order position of the Accumulator entering the sign of the Extended Accumulator. If the count is positive, the content of the Accumulator and Extended Accumulator is shifted to the left the number of positions specified by the count, with bits leaving the sign of the Extended Accumulator entering the low-order position of the Accumulator and bits leaving the sign position of the Accumulator entering the low-order position of the Extended Accumulator.

Registers altered: Accumulator
Extended Accumulator

Timing: 7.5 μs + 1.5 μs per position shifted; if subscripted, 10.5 μs + 1.5 μs per position shifted

NORMALIZED | 0 0 | 0 0 | 1 4 |

The content of the Accumulator and Extended Accumulator is shifted to the left until the 17th and 18th bits of the Accumulator are different. The sign bit of the Extended Accumulator is not shifted. Bits leaving the 17th bit of the Extended Accumulator enter the low-order position of the Accumulator. Zeros fill the positions vacated on the right. A count of the number of positions shifted is retained as a 6-bit positive number in the Scale Register. If the content of the Accumulator and positions 1 through 17 of the Extended Accumulator are zero, the Scale Register is set to zero.

Registers altered: Accumulator
Extended Accumulator
Scale Register

Timing: 6.0 μs + 1.5 μs per position shifted, or 31.5 μs if Accumulator and Extended Accumulator are both zero.

CLOSE EXTENSION WITH DECISION | 0 0 | 0 0 | 1 3 |

The content of the Accumulator and Extended Accumulator is shifted left one position. The sign of the Extended Accumulator is not shifted, and the vacated, low-order position of the Extended Accumulator is filled with the content of the Decision Register. The Overflow Register is not altered.

Registers altered: Accumulator
Extended Accumulator

Timing: 4.5 μs

REVERSE BITS OF THE ACCUMULATOR | 0 0 | 0 0 | 2 2 |

The contents of the Accumulator are reversed in order. The $(19 - n)$th and nth bits are exchanges (for $n = 1, 9$).

Registers altered: Accumulator

Timing: 4.5 μs

Subscript Instructions

USE SUBSCRIPT noun | 5 4 | n o u n |
USE SUBSCRIPT SUBSCRIPTED noun | 5 5 | n o u n |

The content of storage at the effective address is placed in the Subscript Register.

Registers altered: Subscript Register

Timing: 6.0 μs, 9.0 μs if subscripted

SAVE SUBSCRIPT IN noun | 7 4 | n o u n |
SAVE SUBSCRIPT IN SUBSCRIPTED noun | 7 5 | n o u n |

The content of the Subscript Register is stored at the effective address unless that address is protected by the Storage Limit Registers. If storage is protected, no write into memory occurs.

Registers altered: none

Timing: 9.0 μs, 12.0 μs if subscripted

STEP SUBSCRIPT BY noun | 0 2 | n o u n |
STEP SUBSCRIPT BY SUBSCRIPTED noun | 0 3 | n o u n |

The content of storage at the effective address is added to the content of the Subscript Register. The 18-bit result of the two's complement addition is retained in the Subscript Register.

Registers altered: Subscript Register

Timing: 7.5 μs, 10.5 μs if subscripted

Transfer Instructions

THEN GO TO noun	4 2	n o u n
THEN GO TO SUBSCRIPTED noun	4 3	n o u n

If the content of the Decision Register is zero, the next sequential instruction is executed. If the content of the Decision Register is one, the content of storage at the effective address is placed in the Instruction Counter, and execution proceeds from the address specified by the Instruction Counter. The Decision Register and Or/And Register are reset to zero.

Registers altered: Decision Register
Or/And Register

Timing: 6.0 μs, 9.0 μs if subscripted

GO TO noun, RETURN FROM noun	6 2	n o u n
GO TO SUBSCRIPTED noun, RETURN FROM SUBSCRIPTED noun	6 3	n o u n

The content of storage at the effective address is placed in the Instruction Counter, and execution proceeds from the address specified by the Instruction Counter.

Registers altered: none

Timing: 6.0 μs, 9.0 μs if subscripted

TRANSFORMED BY noun, PERFORM noun	0 6	n o u n
TRANSFORMED BY SUBSCRIPTED noun, PERFORM SUBSCRIPTED noun	0 7	n o u n

The content of the Instruction Counter plus one is stored at the effective address unless that address is protected by the Storage Limit Registers. If storage is protected, no write into memory occurs. The content of one location greater than the effective address is placed in the Instruction Counter, and execution proceeds from the address specified by the Instruction Counter.

Registers altered: none

Timing: 12.0 μs, 15.0 μs if subscripted

Test Instructions

AND	0 0	0 0	1 1

The Or/And Register is set to one.

Registers altered: Or/And Register

Timing: 4.5 μs

OR | 0 0 | 0 0 | 1 5 |

The Or/And Register is set to zero.

Registers altered: Or/And Register

Timing: 4.5 μs

IS LESS THAN noun | 2 6 | n o u n |
IS LESS THAN SUBSCRIPTED noun | 2 7 | n o u n |

If the content of the Accumulator is less than the content of storage at the effective address, the test condition is set to one. Otherwise, it is zero. The Or/And Register's being zero specifies that the test condition is to be or-gated with the content of the Decision Register and that the result is to be retained in the Decision Register. The Or/And Register's being one specifies that the test condition is to be and-gated with the content of the Decision Register and that one result is to be retained in the Decision Register.

Registers altered: Decision Register

Timing: 7.5 μs, 10.5 μs if subscripted

IS EQUAL TO noun | 4 6 | n o u n |
IS EQUAL TO SUBSCRIPTED noun | 4 7 | n o u n |

If the content of the Accumulator is equal to the content of storage at the effective address, the test condition is set to 1. Otherwise it is zero. The Or/And Register's being zero specifies that the test condition is to be or-gated with the content of the Decision Register and that the result is to be retained in the Decision Register. The Or/And Register's being one specifies that the test condition is to be and-gated with the content of the Decision Register and that the result is to be retained in the Decision Register.

Registers altered: Decision Register

Timing: 7.5 μs, 10.5 μs if subscripted

IS GREATER THAN noun | 6 6 | n o u n |
IS GREATER THAN SUBSCRIPTED noun | 6 7 | n o u n |

If the content of the Accumulator is greater than the content of storage at the effective address, the test condition is set to one. Otherwise, it is zero. The Or/And Register's being zero specifies that the test condition is to be or-gated with the content of the Decision Register and that the result is to be retained in the Decision Register. The Or/And Register's being one specifies that the test condition is to be and-gated with the content of the Decision Register and that the result is to be retained in the Decision Register.

Registers altered: Decision Register

Timing: 7.5 μs

IF OVERFLOW 0 0|0 0|0 1

If the content of the Overflow Register is one, the test condition is set to one. Otherwise, it is zero. The Overflow Register is reset to zero. The Or/And Register's being zero specifies that the test condition is to be or-gated with the content of the Decision Register and that the result is to be retained in the Decision Register. The Or/And Register's being one specifies that the test condition is to be and-gated with the content of the Decision Register and that the result is to be retained in the Decision Register.

Registers altered: Decision Register
Overflow Register

Timing: 4.5 μs

IF PARITY ODD 0 0|0 0|0 5

If the number of ones in the 18-bit Accumulator is odd, the test condition is set to one. Otherwise, it is zero. The Or/And Register's being zero specifies that the test condition is to be or-gated with the content of the Decision Register and that the result is to be retained in the Decision Register. The Or/And Register's being one specifies that the test condition is to be and-gated with the content of the Decision Register and that the result is to be retained in the Decision Register.

Registers altered: Decision Register

Timing: 31.5 μs

IS POSITIVE 0 0|0 0|0 3

If the sign position (bit 18) of the Accumulator contains a zero, the test condition is set to one. Otherwise, it is zero. The Or/And Register's being zero specifies that the test condition is to be or-gated with the content of the Decision Register and that the result is to be retained in the Decision Register. The Or/And Register's being one specifies that the test condition is to be and-gated with the content of the Decision Register and that the result is to be retained in the Decision Register.

Registers altered: Decision Register

Timing: 4.5 μs

IS EQUAL TO ZERO 0 0|0 0|2 1

If the value of the contents of the Accumulator is equal to zero, the test condition is set to one. Otherwise, it is zero. The Or/And Register's being zero specifies that the test condition is to be or-gated with the content of the Decision Register and that the

result is to be retained in the Decision Register. The Or/And Register's being one specifies that the test condition is to be and-gated with the content of the Decision Register and that the result is to be retained in the Decision Register.

Registers altered: Decision Register

Timing: 4.5 μs

IF SUBSCRIPT IS NOT GREATER THAN noun 22 noun
IF SUBSCRIPT IS NOT GREATER THAN SUBSCRIPTED noun 23 noun

If the content of the Subscript Register is less than or equal to the content of storage at the effective address, the test condition is set to one. Otherwise, it is zero. The Or/And Register's being zero specifies that the test condition is to be or-gated with the content of the Decision Register and that the result is to be retained in the Decision Register. The Or/And Register's being one specifies that the test condition is to be and-gated with the content of the Decision Register and that the result is to be retained in the Decision Register.

Registers altered: Decision Register

Timing: 7.5 μs, 10.5 μs if subscripted

IS FALSE 00 00 17

The Decision Register is complemented.

Registers altered: Decision Register

Timing: 4.5 μs

Control Instructions

PASS 00 00 02

No operation is performed other than the automatic incrementing of the Instruction Counter.

Registers altered: none

Timing: 4.5 μs

HALT 00 00 00

The processor stops indefinitely. An initiate signal must be supplied from an external source to start the processor.

Registers altered: none

Timing: none

EXECUTE noun	1 2 | n o u n
EXECUTE SUBSCRIPTED noun	1 3 | n o u n

The content of storage at the effective address is used as the address of the instruction to be executed. The Instruction Counter is incremented by one unless it is changed by the execution of a transfer-type instruction. If the machine attempts to execute an *EXECUTE noun*, the program proceeds with no operation being performed.

Timing: 6.0 μs

Miscellaneous Register Instructions

SET SCALE WITH noun	3 2 | n o u n
SET SCALE WITH SUBSCRIPTED noun	3 3 | n o u n

The 6 low-order bits of the content of storage at the effective address are placed in the Scale Register.

Registers altered: Scale Register

Timing: 6.0 μs, 9.0 μs if subscripted

LET SCALE 0 0 | 0 0 | 2 0

The content of the Scale Register is placed in the 6 low-order bits of the Accumulator. The 12 high-order bits of the Accumulator are set to zero.

Registers altered: Accumulator

Timing 4.5 μs

SET PAGE 0 0 | 0 0 | 1 2

The content of bits 13 through 16 of the Accumulator are placed in the Page Register.

Registers altered: Page Register

Timing: 4.5 μs

RESET OVERFLOW 0 0 | 0 0 | 0 7

The content of the Overflow Register is set to zero.

Registers altered: Overflow Register

Timing: 4.5 μs

RESET DECISION 0 0 | 0 0 | 2 3

The content of the Decision Register is set to zero.

Registers altered: Decision Register

Timing: 4.5 μs

EXIT | 0 0 | 0 0 | 1 6 |

This instruction initiates interrupt number 16 and uses locations octal 200 through 207. Upon completion, execution proceeds normally, using the new value in the Instruction Counter.

Registers altered: Limit Register
Interrupt Priority Register
Page Register
Or/And Register
Overflow Register
Carry Register
Decision Register
Scale Register

Timing: 42.0 μs

RESUME FROM noun | 7 2 | n o u n |
RESUME FROM SUBSCRIPTED noun | 7 3 | n o u n |

The content of storage at the effective address is used as the starting address of an interrupt storage area. This instruction reloads the registers that were saved at the occurrence of an interrupt. Upon completion, execution proceeds normally, using the new value in the Instruction Counter.

Registers altered: Limit Register
Interrupt Priority Register
Page Register
Or/And Register
Overflow Register
Carry Register
Decision Register
Scale Register

Timing: 25.5 μs, 28.5 μs if subscripted

Input/Output Instruction

CONNECT TO noun | 1 6 | n o u n |
CONNECT TO SUBSCRIPTED noun | 1 7 | n o u n |

The content of storage at the effective address is sent to the I/O unit as a command. This word must have a zero in bits 17 and 18 to denote the establishment of a cycle-steal channel. Bits 1 through 16 specify the data channel and block length. The content of the Accumulator is stored at location 7. The content of location 7 is then output as a starting memory address to the I/O unit.

Registers altered: none

Timing: 13.5 μs, 16.5 μs if subscripted

LET FUNCTION TO noun		1 6	n o u n
LET FUNCTION TO SUBSCRIPTED noun		1 7	n o u n

The content of storage at the effective address is sent to the I/O unit as a command. This word must have a zero in bit 18 and a one in bit 17 to denote that bits 1 through 16 are a function code.

Registers altered: none

Timing: 6.0 μs, 9.0 μs if subscripted

OUTPUT TO noun		1 6	n o u n
OUTPUT TO SUBSCRIPTED noun		1 7	n o u n

The content of storage at the effective address is sent to the I/O unit as a command. This word must have a one in bit 18 and a zero in bit 17. Bits 1 through 16 denote the data channel for an output device. The content of the Accumulator, stored at location 7, is then output as data to the I/O unit.

Registers altered: none

Timing: 13.5 μs, 16.5 μs if subscripted

LET INPUT FROM noun		1 6	n o u n
LET INPUT FROM SUBSCRIPTED noun		1 7	n o u n

The content of storage at the effective address is sent to the I/O unit as a command. This word must have a one in bits 17 and 18. Bits 1 through 16 denote the data channel for an input device. The I/O unit stores one word of data at location 7. The content of location 7 is then placed in the Accumulator.

Registers altered: Accumulator

Timing: 13.5 μs, 16.5 μs if subscripted

INTERRUPT PROCESSING

Upon completion of each instruction, the processor senses if any of the 15 external interrupts appeared during the execution of the previous instruction (eight interrupts are possible on the OBP). The order of hardware priority runs from interrupt 1 as the highest to interrupt 8 as the lowest. Hardware priority is used only to prevent conflict when two or more interrupts allowed by priority status are waiting to be processed. The lockout status register in the I/O unit provides programming control of the order in which interrupts are processed.

The priority status word, as shown in Figure C4, comprises a data transfer from the memory to the I/O unit in which bits 1 through 8 correspond to the 8 interrupt levels. The programmer determines the bit pattern contained in the interrupt location and thereby sets the priority status.

When an interrupt arrives at the I/O unit on one of the 8 interrupt lines, it is compared with the corresponding bit position of the priority status register. If the current priority status has a one in the respective bit position for that interrupt, the interrupt is temporarily locked out and cannot get to the processor. A flip-flop in the I/O unit holds the interrupt so that if at some later time the priority status is changed and a zero appears in the corresponding bit position of the saved interrupt, the interrupt will be honored and allowed to pass on to the processor. When the processor receives the interrupt, an acknowledge signal is sent back to the I/O unit, resetting the flip-flop so that another low-priority interrupt may be saved. Interrupts arriving at the I/O unit on a channel that is currently holding a previous interrupt naturally will be lost. The priority status is altered when an interrupt, an *EXIT* instruction, or a *RESUME FROM* instruction is processed by the processor.

When an interrupt is honored (for example, interrupt n, as illustrated by Figure C5), the following registers are stored into and loaded from the indicated interrupt locations.

Interrupt priority status	Location 8 X n (bits 1 to 16)
Scale Register	Location 8 X (n + 1) (bits 1 to 6)
Carry Register	Location 8 X (n + 1) (bit 7)
Overflow Register	Location 8 X (n + 1) (bit 8)
Or/And Register	Location 8 X (n + 1) (bit 9)
Decision Register	Location 8 X (n + 1) (bit 10)
Page Register	Location 8 X (n + 2) (bits 13 to 16)
Storage Limit Register	Location 8 X (n + 2) (bits 1 to 18)
Instruction Counter	Location 8 X (n + 3) (bits 1 to 16)
Location 8 X (n + 4) (bits 1 to 16)	Interrupt priority status
Location 8 X (n + 5) (bits 1 to 6)	Scale Register
Location 8 X (n + 5) (bit 7)	Carry Register
Location 8 X (n + 5) (bit 8)	Overflow Register
Location 8 X (n + 5) (bit 9)	Or/And Register
Location 8 X (n + 5) (bit 10)	Decision Register
Location 8 X (n + 5) (bits 13 to 16)	Page Register
Location 8 X (n + 6) (bits 1 to 18)	Storage Limit Register
Location 8 X (n + 7) (bits 1 to 16)	Instruction Counter

(Upon loading of the Instruction Counter, execution proceeds using the new value.)

When the *RESUME FROM noun* instruction is executed (the user is responsible for directing the assembler to preset *noun* with the value 8 X n), the following registers are restored.

Location 8 X n (bits 1 to 16)	Interrupt priority status
Location 8 X (n + 1) (bits 1 to 6)	Scale Register

Location 8 X (n + 1) (bit 7)	Carry Register
Location 8 X (n + 1) (bit 8)	Overflow Register
Location 8 X (n + 1) (bit 9)	Or/And Register
Location 8 X (n + 1) (bit 10)	Decision Register
Location 8 X (n + 1) (bits 13 to 16)	Page Register
Location 8 X (n + 2) (bits 1 to 18)	Storage Limit Register
Location 8 X (n + 3) (bits 1 to 16)	Instruction Counter

(Upon completion, execution begins at the new value of the Instruction Counter.)

For an interrupt to function properly, the interrupt locations 8 X (n + 4), 8 X (n + 5), 8 X (n + 6), and 8 X (n + 7) must be preset to the appropriate values during the assembly of the user's program. Normally the Page, Decision, Or/And, Overflow, and Carry Registers will all be zero; therefore, location 8 X (n + 5) may be preset to the desired Scale Register setting. Location 8 X (n + 7) must contain the instruction address to which the program is to be transferred when the nth interrupt is honored.

The *EXIT* instruction performs exactly like an interrupt with n = 16.

STORAGE LIMIT CHECKING

The purpose of the SLR is to allow the writing of data into memory only in those areas assigned to the program. This provides storage protection which prevents a program from writing into any code area or from accidentally writing into someone else's data area.

The protection is dynamic in that it can be modified by the execution of either of two privileged instructions (reserved for the *EXEC* program) or by the execution of any external interrupt.

The SLR is divided into two 9-bit halves which represent an upper limit and lower limit, as shown in Figure C6. The nine most significant bits of the Address Register (ADR) are compared by special logic with the corresponding upper or lower limits, and both tests indicated in the figure must be positive for a write to be allowed. Note that storage may be completely turned off for a program if the lower limit is greater than the upper limit.

Since the six least significant bits of the ADR are not checked, they are "don't care" bits; hence, the smallest area of definition is 128 words. Any block size in multiples of 128 words may be assigned to a program (the OBP-1 16K memory contains a total of 128 blocks). The memory may be mapped into either exclusive or overlapping data blocks in any desired pattern. As in other areas, the CPU is fully implemented for a 64K memory size; therefore, bits 15 and 16 of the address are used for storage limit checking, although they serve no other purpose with the OBP-1 16K memory size.

The instructions that cause a memory write and which are checked for storage limits are *YIELD, TRANSFORMED BY, SAVE EXTENSION IN,* and *SAVE SUBSCRIPT IN.* Note also that the ADR at the time of the storage limit test contains the effective address, which means that the subscript register value is included if these instructions should be subscripted.

If a write is not permitted because of failure of the storage limit test, a read cycle will actually occur; however, more importantly, at the same time an external interrupt (called *OUTLIMIT*) will be generated. This causes a diagnostic routine to be entered which will at least identify the culprit (from its last IC) and possibly turn it off.

SCALING AND OVERFLOW

Scaling

The arithmetic unit of the OBP normally regards all numbers as fractional. The Scale Register, however, permits the user to consider numbers as integer, fractional, or mixed, because the true value of the number contained in the Accumulator may be considered to be $2^s(\hat{n})$, where s is the value of the number contained in the Scale Register and $\hat{n}$ is the fractional value of the number in the Accumulator. This multiplication is equivalent to a left or right shift of $\hat{n}$ by s places depending on s being positive or negative, respectively.

The Scale Register is a 6-bit register that indicates the position of the binary point of the number contained in the Accumulator. Negative values of the Scale Register are represented in two's complement form, and the uppermost bit of this register is the sign bit. Thus, there are five bits available in the Scale Register to contain a maximum positive value of $2^5 - 1$ (+31) or a maximum negative value of -2^5 (−32). Therefore, the Scale Register allows the OBP to process numbers in the range of 2^{-32} through 2^{+31}. The size of the Accumulator, however, permits only 17 bits of significance to be maintained. Numbers that have lost all significance appear as zero.

In addition to indicating the scaling of the content of the Accumulator, the Scale Register automatically subjects the fractional product and dividend to a shift of length s, where s is the value contained in the Scale Register. If the value of s is zero, no shift takes place, and the final result is truly fractional. This is known as the fractional mode, with the binary point being immediately to the right of the sign bit. A Scale Register setting of 17 is called the integer mode, with the binary point being to the right of the least significant bit of the Accumulator.

PLUS, MINUS, NEGATE, and *PLUS CARRY* instructions are all independent of the setting of the Scale Register. The programmer is responsible for obtaining the binary point position of the number in the Accumulator as well as the operand from storage.

In Figures C7 and C8, several examples of scaling, with various values contained in the Scale Register and the subsequent location of the binary point, are given for multiplication and division, respectively. The operations of multiplication and division on A and B can be described symbolically in terms of the fractional values the hardware sees ($\hat{A}$ and $\hat{B}$) and the fixed Scale Register setting (s):

(1) Multiplication:

$$A \cdot B = 2^s\hat{A} \cdot 2^s\hat{B} = 2^{2s}(\hat{A} \cdot \hat{B}) = 2^s\hat{C}.$$

Thus, the fractional result obtained by the hardware is

$$\hat{C} = 2^s(\hat{A} \cdot \hat{B}) .$$

This explains the double-length shift by the value of the Scale Register subsequent to multiplication.

(2) Division

$$A/B = 2^s\hat{A}/2^s\hat{B} = \hat{A}/\hat{B} = 2^s\hat{C} .$$

Now,

$$\hat{C} = 2^{-s}\hat{A}/\hat{B} .$$

This explains the double-length shift by the value of the two's complement of the Scale Register prior to division.

Overflow and Carry

Overflow, a condition that occurs whenever there is a loss of a significant bit out of the 17th bit position of the Accumulator, is indicative of an erroneous computation due either to improper scaling or to too large a number being used or generated. Since all numbers are regarded as fractional during the performance of an arithmetic operation, the condition of overflow is equivalent to obtaining a result that is outside the permitted fractional number range. In general, overflow will cause the sign of the result to be incorrect. An overflow will occur–

(1) On a *PLUS* or *PLUS CARRY* instruction only if the signs of the two numbers are alike and the magnitude of the sum is greater than $1.0 - 2^{-17}$.

(2) On a *MINUS* instruction only if the signs differ and the magnitude of the difference is greater than $1.0 - 2^{-17}$.

(3) On a *NEGATE* instruction only when bits 1 through 17 are all zero (i.e., zero, or -1.0).

(4) Multiplication of two fractional numbers produces a fractional number that is less than the magnitude of the larger number. Thus, overflow cannot occur during multiplication with the Scale Register set less than or equal to zero. An exception is multiplication of -1.0 by -1.0 with a Scale Register setting of zero. This results in a product of -1.0 and an overflow indication. Positive Scale Register settings may cause overflow when the product is shifted.

(5) Division of a number by an equal or smaller number yields a fractional result greater than or equal to one, thus producing overflow. This condition can be avoided by proper shifting of the numbers involved. Negative Scale Register settings may cause overflow when the dividend is shifted prior to fractional division.

(6) Shifting to the right cannot cause overflow since the magnitude of the number decreases. However, overflow can occur on a shift or double shift to the left if the sign of the Accumulator is changed during the shifting process.

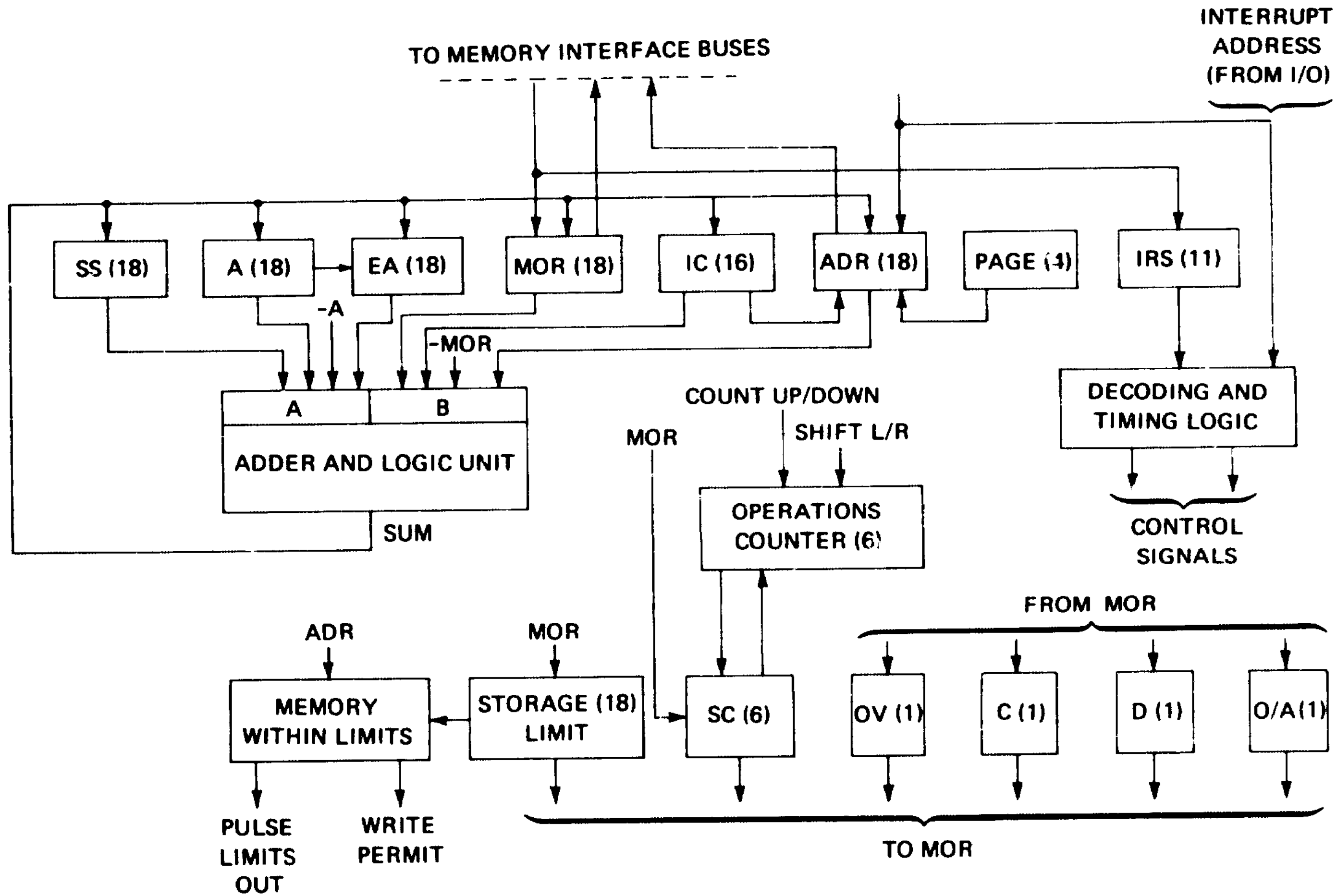


Figure C1—CPU block diagram.

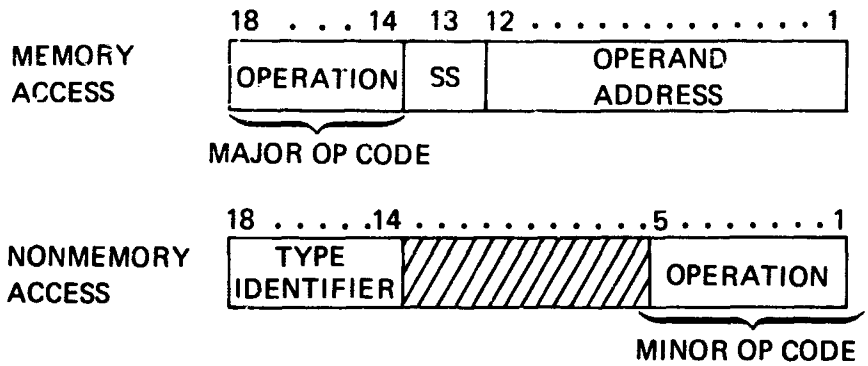


Figure C2—Instruction word.

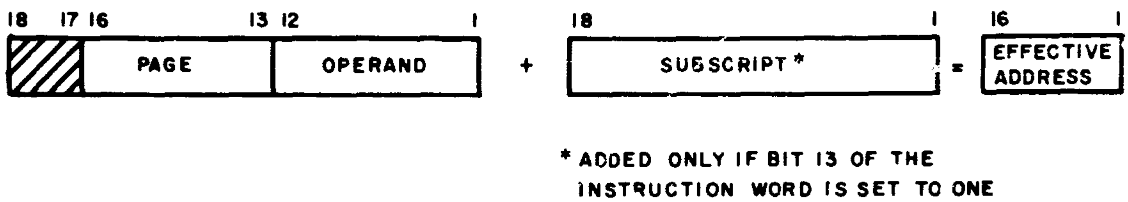


Figure C3—Effective address.

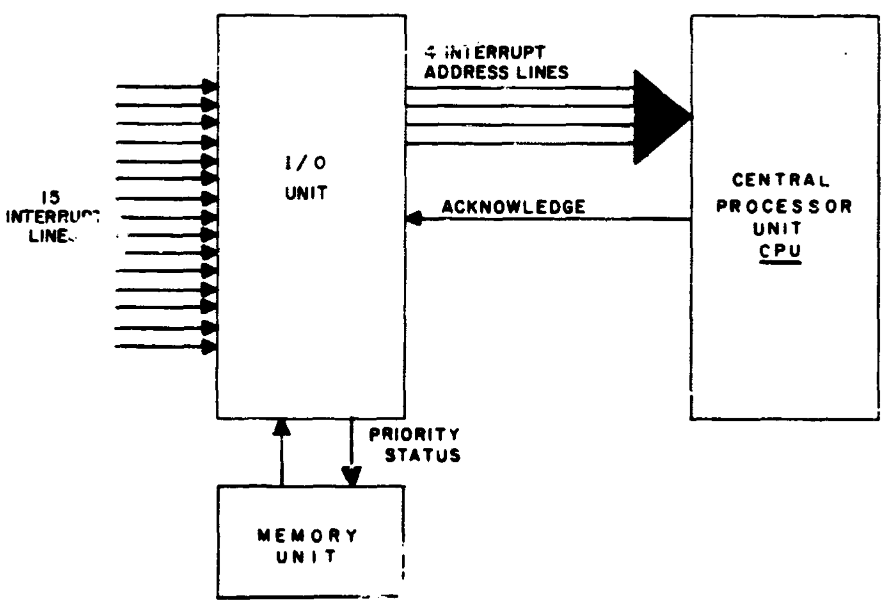


Figure C4—OBI interrupt block diagram.

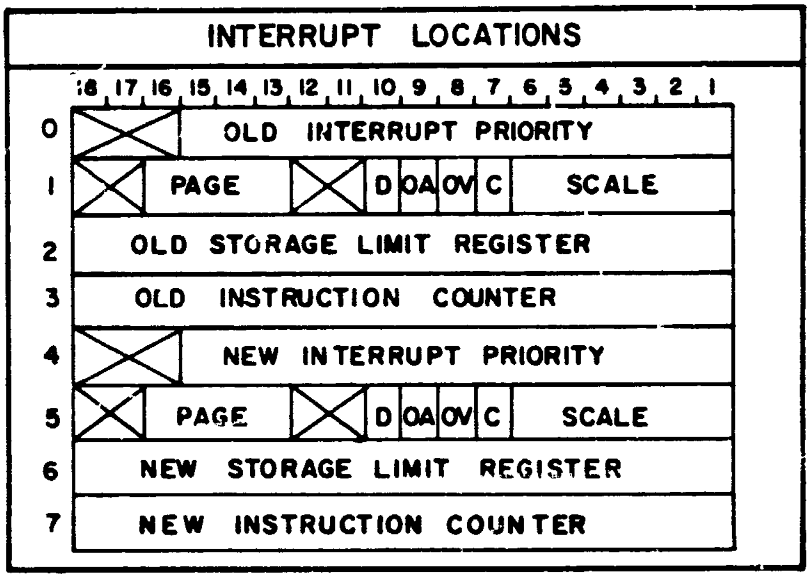


Figure C5—Interrupt locations.

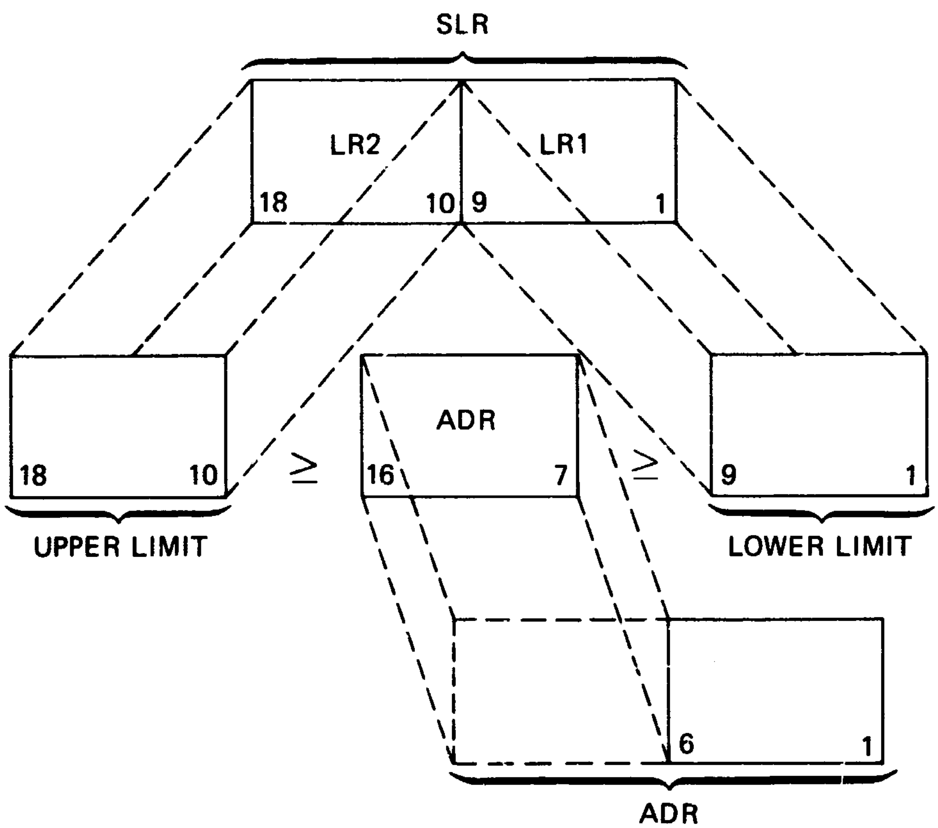


Figure C6—Storage limit test.

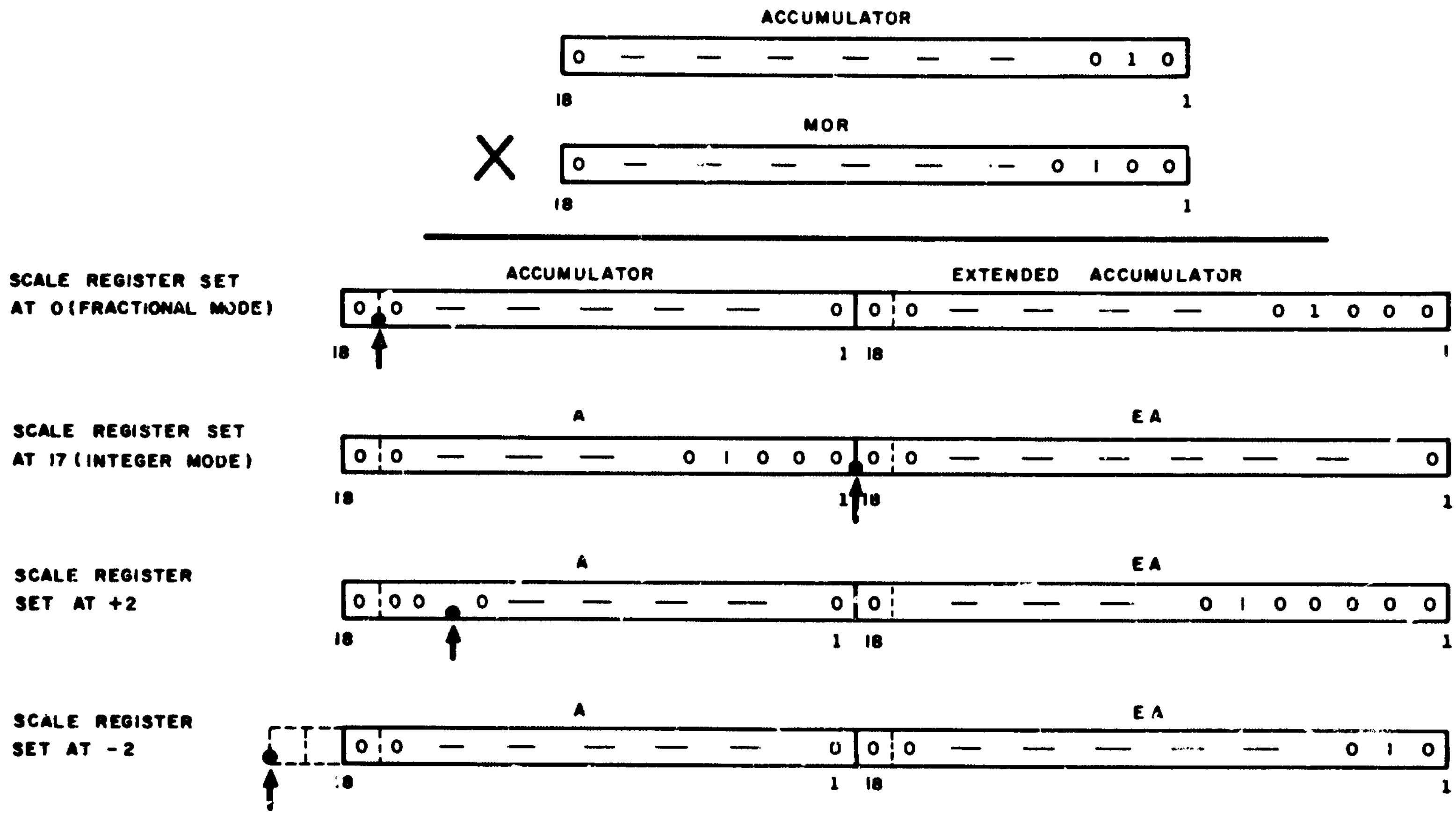


Figure C7—Binary point location in multiplication with scale registei settings of 0, 17, +2, and −2.

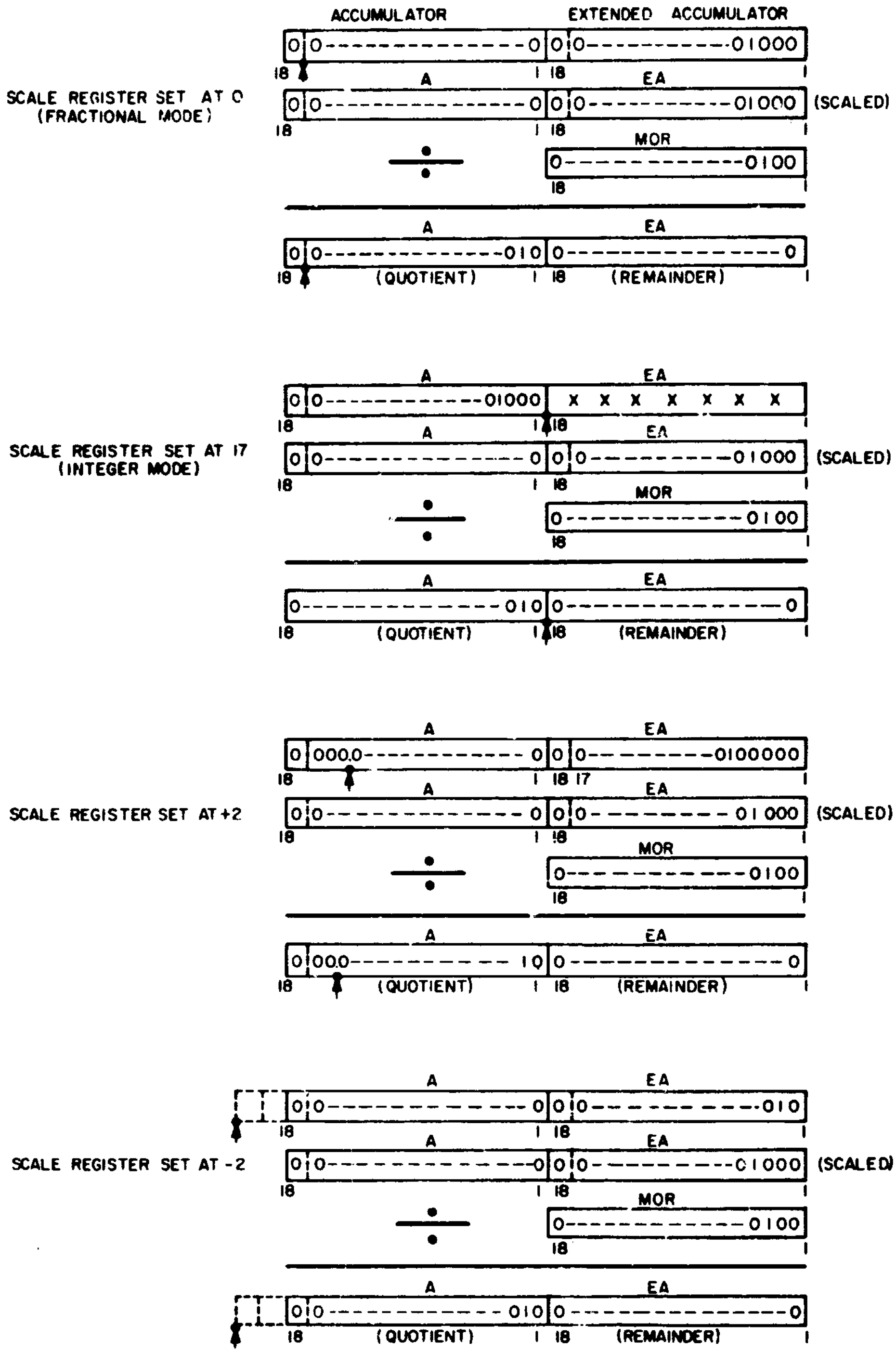


Figure C8—Binary point location in division with scale register settings of 0, 17, +2, and −2.

PRECEDING PAGE BLANK NOT FILMED

APPENDIX D

QUALIFICATION/ACCEPTANCE TEST PLAN*

This appendix outlines the environmental and functional tests performed to qualify and accept the OBP for use in the OAO C spacecraft.

The purpose of the test is to demonstrate the structural strength and functional performance of the OBP in accordance with NASA Specification SO-S345-1 and revisions thereof.

The OBP is a general-purpose, stored-program digital computer with capabilities similar to the SDS-920 or SDS-930 ground systems computers. It contains a 16 384-word, 18-bit-per-word core memory in the form of four modules of 4096 words each. The two largest modules are the CPU and the I/O unit, with the latter containing all necessary interface hardware to connect the OBP to the OAO command and data handling, power, and stabilization and control subsystems. The remaining module is the power converter, which is mounted in bay G2 separate from the rest of the system, which resides in bay G1 immediately above. Maximum weight of the OBP is 64 lb.

For radio frequency interference and thermal vacuum testing, the test specimen consists of seven flight boxes: Six (4 memories, the I/O unit, and the CPU) are mounted on a single plate, bay G1 door, or equivalent; and the remaining package (the power converter) is mounted on a separate panel simulating bay G2 door.

For vibration and shock testing, the test specimen consists of three independent unpowered modules (the I/O unit, the CPU, and the power converter) which will be tested as individual "black boxes". Memories will be tested separately at the vendor according to Electronic Memories Inc. Test Plan TA920269.**

Applicable documents are–

NASA Specification SO-S345-1
Electronic Memories Inc. Test Plan TA920269**
OBP 2 Functional Test Procedure
EMI/RFI Test Procedure for the OBP 2

*Excerpted from "OAO C Onboard Processor Qualification/Acceptance Test Plan", an in-house document by the Flight Data Storage Branch, GSFC.

**Electronic Memories Inc., 12621 Chadron Ave., Hawthorne, California.

TEST AND FAILURE REPORTING

Failure Report

A failure report for each component failure encountered in acceptance testing shall be prepared in accordance with the OAO failure-reporting format. A failure is defined as any reading that is not within the tolerances specified by the functional test procedure when the inputs are within limits. All failures shall be reported to the OAO Project Manager or his representative and shall be fully documented. A decision as to whether the test shall continue following a failure will be made by the Cognizant Engineer and/or the OAO Cognizant Environmental Engineer.

Failure Analysis

All failures and any erratic performance even within the specification limits will be investigated, and a proposal for corrective action will be submitted to the OAO Environmental Committee for review. A failure analysis will be conducted when the failure is considered sufficiently significant to require action. The Failure Analysis Review will be performed by the Design Engineer, Systems Manager, Quality Control Engineer, Spacecraft Support Manager, and Subsystem Engineer.

Test Summary

Within 5 working days after completion of the component acceptance, a test summary will be prepared which will contain the following:

(1) Summary of tests performed.

(2) Description of any deviation in test procedure, together with documentation of authority for change.

(3) Cause of failure (if a failure occurred), results of investigations, and remedial steps taken.

(4) Sample data sheet summarizing the major performance characteristics of the unit.

(5) List of all out-of-specification measurements requiring waiver action.

(6) Vibration plots.

Test Report

Within 30 days after completion of component testing, a test report will be prepared which will contain the following:

(1) Summary of results and conclusions.

(2) Description of test performed.

(3) Test conditions.

(4) List of test equipment used and calibration data.

(5) Any test peculiarities or deviations.

(6) Reduced data and graphs.

(7) Raw data sheets.

SUMMARY OF TESTS

Visual inspection will be performed after fabrication and after the OBP has been subjected to each environmental condition.*

A functional test will be performed on the OBP prior to commencement of environmental test, upon completion of vibration and shock tests, and during thermal vacuum test.

Environmental tests shall be performed on the OBP in the following sequence:

(1) Baseline functional.

(2) EMI/RFI.

(3) Vibration, sinusoidal.**

(4) Vibration, random.**

(5) Shock.**

(6) Thermal vacuum.

TEST CONDITIONS AND EQUIPMENT

Unless otherwise specified, tests shall be conducted under the following conditions:

Temperature	74 ±10°F (23 ±5°C)
Relative humidity	75 percent or less
Barometric pressure	28 to 32 in. Hg

Electromagnetic and radio frequency interference test equipment shall include the following:

(1) OAO Experimenter's Test Control Unit (ETCU).

(2) SDS-930 computer.

(3) EMI/RFI test equipment as specified in the EMI/RFI Test Procedure for the OBP 2.

Thermal vacuum test equipment shall include the following:

(1) ETCU.

*Bolt torques to be checked prior to thermal vacuum.
**These sequences may be completed on a per axis basis.

(2) SDS-930 computer.

(3) OBP test racks (2).

Vibration test equipment shall include the following:

(1) Multimeter (Simpson Model 260).

(2) Polaroid camera.

(3) Accelerometers as required.

Instruments requiring calibration shall have stickers affixed showing calibration dates no more than 6 months old upon completion of the qualification test.

TEST DESCRIPTION

Prior to commencement of tests, the OBP shall be inspected for conformance to the applicable GSFC drawing. In addition, the OBP shall be inspected for workmanship, finish, and any discernible defects. Visual inspection shall be repeated after each environmental test.

The OBP "flight" unit will be connected to the Code 562 computer/spacecraft interface equipment and subjected to typical orbital operations, including the exercise of all modes of operation as specified in functional test procedure.

Electromagnetic and radio frequency interference tests shall be conducted in accordance with the EMI/RFI Test Procedure for the OBP 2.

Vibration tests will be conducted on three individual units of the OBP system as black boxes. The mounting hole pattern is identical for all three of these units. The CPU and I/O unit will weigh approximately 13 lb; the power converter will weigh about 8 lb. There will be no wires or cables connected to any of the units. A limited amount of electrical checking may be conducted between axes only. Complete functional tests will be run both before and after the vibration test.

Vibration test tolerances will be as follows: Sinusoidal vibration amplitude, ±10 percent. Vibration frequency, ±2 percent. Random vibration spectral density, ±3 dB.*

The component shall be subjected to sinusoidal vibration in each of the three mutually perpendicular axes shown in Figure D1. The amplitude shall be varied with frequency as shown in Table D1 at the rate of 4 octaves per minute. Four control accelerometers shall be mounted on the vibration fixture, one adjacent to each corner. The average shall be used for servo control. An xy-plot of the control level acceleration shall be provided. A 1-g low level sweep shall be performed for each axis prior to the full level sweep.

The component shall be subjected to random vibration over a frequency range of 15 to 2000 Hz at the levels given in Table D1 for a period of 2 minutes per axis. An equalization

*+10% total g rms; −5% total g rms.

Table D1—Spacecraft acceptance levels.

Sinusoidal All Axes (4 octaves per minute)		Random All Axes (2 minutes per axis)	
Frequency (Hz)	Level	Frequency (Hz)	Level
5 to 20	1/2 in. displacement	15	0.010 g^2/Hz
20 to 110	10.0-g peak	15 to 70	linear increase
110 to 2000	5.0-g peak	70 to 100	0.31 g^2/Hz
		100 to 400	linear decrease
		400 to 2000	0.02 g^2/Hz

plot shall be recorded for each axis. An xy-plot of the control level acceleration shall be provided.

The shock pulse shall approximate a half-sine pulse between 10-percent amplitude points as follows: amplitude tolerance, ±15 percent of peak g force; duration, ±20 percent. Four shock pulses will be administered along each axis according to the following schedule: x-axis (thrust)—two shocks (±) 30g for 6-ms duration and two shocks (±) 30g for 12-ms duration; y-axis (lateral)—two shocks (±) 15g for 6-ms duration and two shocks (±) 15g for 12-ms duration; and z-axis (lateral)—same as y-axis. (May be completed during vibration testing if expedient.)

The mounting plates of the OBP and power converter will serve as the control surfaces and will be heated by thermal pads placed in contact with the unused sides. The exposed surfaces shall be covered with 25 layers of spacecraft insulation. Thermocouple temperatures shall be recorded at 30-minute intervals, and the chamber pressure shall be recorded at 30-minute intervals.

The components, with power applied, shall be placed in a temperature-vacuum chamber and subjected to a vacuum of 10^{-6} mm Hg. The heat-sink temperatures will be allowed to stabilize at 80° ±5°F, as indicated by an average of the thermocouple readings. After an hour at this temperature, the functional tests shall be performed. After 4 hours at 80°F, the component heat sink temperatures shall be raised to 150° ±5°F at a maximum rate of 3°F per minute. After 1 hour of temperature stabilization, the functional tests shall be performed. After 8 to 16 hours of temperature stabilization, the functional tests shall be repeated. The total length of time at 150°F shall be no less than 24 hours, with at least one additional repeat of the functional test performed within 8 hours from the end of this period. The heat sink temperatures shall be lowered to −30° ±5°F at a maximum rate of 3°F per minute. After 1 hour of temperature stabilization, the functional tests shall be performed. After 8 to 16 hours of temperature stabilization, the functional tests shall be repeated. The total length of time at −30°F shall be no less than 24 hours, with at least one

additional repeat of the functional test performed within 8 hours from the end of this period. Component heat-sink temperatures shall be raised to 80° ±5°F at a maximum rate of 30°F per minute. The functional tests shall be performed. The chamber shall be returned to ambient conditions and the functional tests shall be performed. This completes the test.

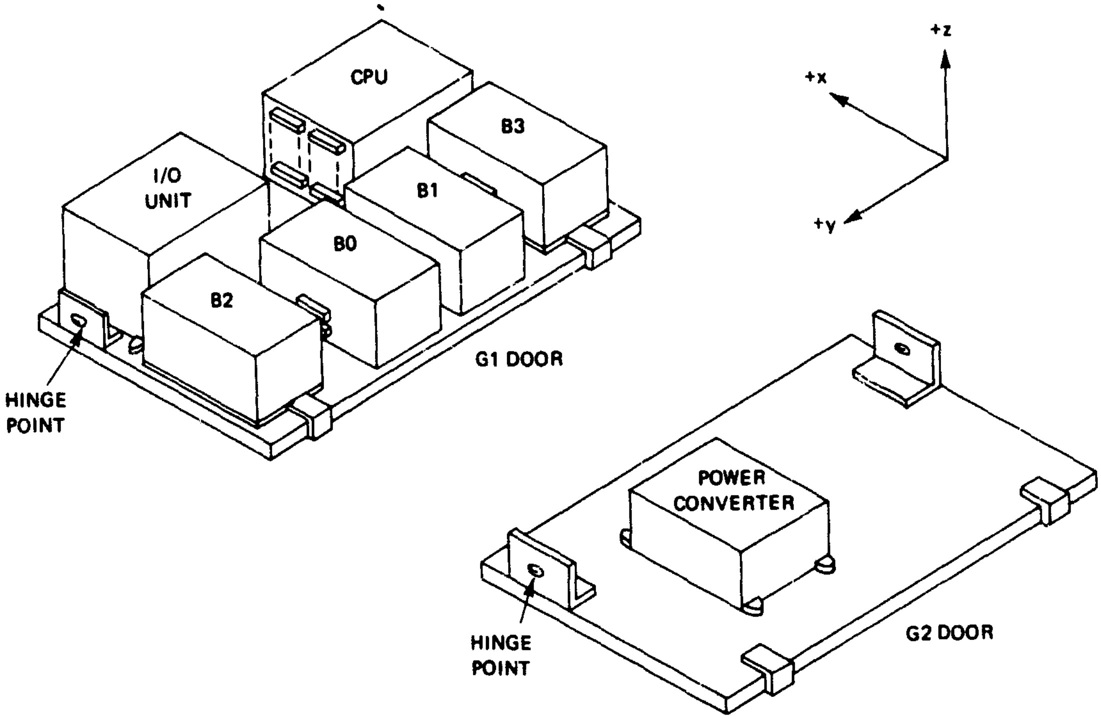


Figure D1—OBP mounting axis orientation.